INSTITUT FÜR INFORMATIK
der Ludwig-Maximilians-Universität München

Increasing user engagement to foster stress awareness using smart watches

Nikolai Fischer

Master Thesis

Aufgabensteller	Prof. Dr. François Bry
Betreuer	Prof. Dr. François Bry, Yingding Wang
Abgabe am	18. September 2018

Erklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit selbständig verfasst habe und keine anderen als die angegebenen Hilfsmittel verwendet habe.

München, den 18. September 2018 Nikolai Fischer

Abstract

Stress can have considerable effects on the human body and mind. While often described as purely negative, effects of stress can also be positive. Awareness of the intensity and the nature of stress can help to live a healthier life while simultaneously taking advantage of the positive effects of stress.

The objective of this work is to improve a smartphone application for stress tracking by adding persuasive and behavior changing features that increase user engagement and thus foster stress awareness. The frequent recording of the user's daily activities and feelings was identified as the main way to achieve this.

A set of characteristics for the mobile application and the addition of a companion smart watch application are proposed to change user behavior towards more engagement and a healthier handling of stress. The application's characteristics stem from approaches from multiple theoretical frameworks, namely Fogg's Behavior Model and Functional Triad for persuasive technology as well as various techniques to enhance the user experience. Chances of wearable devices are also explored in connection with user engagement and behavior change. The motives, the design and the implementation behind the added features are explained in detail. An optimized version of the application is consequently compared to a non-optimized version to evaluate the achievement of the goal. The 43 users of the non-optimized version used the application with a Fitbit fitness tracker, while the 43 users of the persuasive version used a Wear OS smart watch. The study participants used the application for 15 days. During this time their usage was monitored with software analysis tools. After the study the participants responded to a questionnaire which further explored their usage and the effects on their stress awareness.

The optimized, persuasive version of the application was able to increase the number of recorded activities, which was identified as the main user engagement metric. This confirms the successfulness of the persuasive improvements to the application. The higher user engagement resulted in a increased subjective stress awareness in users of the improved application. The study identifies the addition of tunneling and reduction features to the application as most responsible for the increase in user engagement. The thesis concludes with a discussion of the study's limitations and implication for designing persuasive software for wearables and smartphones.

Zusammenfassung

Stress kann erhebliche Auswirkungen auf den menschlichen Körper und Geist haben. Obwohl Stress häufig als rein negativ beschrieben wird, können seine Effekte auch positiv sein. Ein Bewusstsein über die Intensität und die Art von Stress kann dabei helfen, ein gesünderes Leben zu führen und gleichzeitig die positiven Effekte besser auszunutzen.

Das Ziel dieser Arbeit ist es, eine Smartphone Anwendung zur Stressüberwachung um persuasive und verhaltensändernde Eigenschaften zu erweitern. Diese Eigenschaften verstärken die Nutzerbindung (user engagement) und pflegen somit das Stressbewusstsein. Das regelmäßige Aufzeichnen von täglichen Aktivitäten und der damit verbundenen Gefühle durch den Nutzer wurde als die wichtigste Methode identifiziert, das oben beschriebene Ziel zu erreichen.

In der vorliegenden Arbeit werden eine Reihe von Charakteristiken für die mobile Anwendung und das Hinzufügen einer begleitenden Smart Watch Anwendung vorgeschlagen. Diese sollen das Nutzerverhalten hin zu einer stärkeren Nutzerbindung und einer gesünderen Handhabung von Stress verändern. Die Merkmale der Anwendung stammen von Ansätzen verschiedener theoretischer Gerüste ab. Hierzu zählen unter anderem Foggs Behavior Model und Functional Triad für persuasive Technologien sowie verschiedene Techniken, um die Benutzererfahrung zu verbessern. Ebenfalls werden die Möglichkeiten von tragbaren Geräten in Verbindung mit der Nutzerbindung und Verhaltensänderungen erforscht. Das Design, die Implementierung und die Motive hinter den hinzugefügten Funktionen werden im Detail erklärt. Danach wird eine optimierte Version der Anwendung mit einer nicht-optimierten Version verglichen, um die Erreichung des Ziels dieser Arbeit zu überprüfen. Die 43 Anwender der nicht-optimierten Version benutzten die Applikation mit einem Fitbit Fitness Armband, während die 43 Benutzer der persuasiven Version eine Wear OS Smart Watch verwendeten. Die Studienteilnehmer testeten die Anwendung für 15 Tage. Während dieser Zeit wurde ihre Nutzung durch Software Werkzeuge überwacht. Nach der Studie beantworteten die Teilnehmer einen Fragebogen, der ihre Nutzung und die Effekte auf ihre Stresswahrnehmung weiter erforschte.

Die optimierte, persuasive Version der Anwendung konnte die Anzahl der aufgezeichneten Aktivitäten erhöhen, was als Hauptmetrik für die Nutzerbindung identifiziert wurde. Dies bestätigt den Erfolg der persuasiven Verbesserungen der Anwendung. Die höhere Nutzerbindung löste eine gesteigerte subjektive Stresswahrnehmung bei Benutzern der verbesserten Anwendung aus. Die Studie identifiziert das Hinzufügen von Tunneln und Reduktionsfunktionen als bedeutendsten Faktor für den Anstieg der Nutzerbindung. Diese Arbeit endet mit einer Diskussion der Einschränkungen der Studie sowie der Auswirkungen auf den Designprozess von persuasiver Software für tragbare Geräte und Smartphones.

Acknowledgments

I would like to thank Dr François Bry who gave me the opportunity to write this thesis and provided valuable feedback and funding for my studies. I am truly grateful for his support and advice during our meetings.

My sincere thanks also goes to my supervisor M. Sc. Yingding Wang. His guidance, encouragement, and ideas helped me while researching and writing this thesis. He allowed the thesis to be my own, but steered me in the right direction when I needed it. He was always available via email or personally to give me feedback on my writing or help me with technical problems.

Lastly, I also would like to thank everyone who participated in the user study. Their interest and motivation to help this project still amazes me. I would like to express my gratitude for sharing their precious time.

Contents

CHAPTER 1

Introduction

1.1 Background

Eustress and Distress

Stress is considered a "modern ailment" of our society. Many mental and physical diseases can partly be traced back to high stress levels [49]. Stress is not only experienced by the working population but is also present in the lives of university students. These often deal with a high workload originating from courses and assignments. Selye defined stress as "the nonspecific response of the body to any demand made upon it" and also differentiated between positive and negative facets of stress [49]. Positive stress is referred to as **eustress** and describes stress which is beneficiary to a person. **Distress** is the opposite and results in negative reactions by the mind and the body. In a university setting an example for eustress would be a student's slightly increased stress level while she takes an examination. Stress helps her to concentrate better and to improve her performance. Distress, on the other hand, would be a paralyzing fear of the examination resulting in a worse performance while learning or even a complete black-out while taking an examination.

Project Stila

To combat the negative effects of stress in a university environment, a group at Ludwig-Maximilian University of Munich has started the project *Stila*. Stila relies on a heart rate tracking device, an Android application, and a back-end server. The heart rate tracking device is worn by the users to measure their heart rates periodically. In the current form fitness trackers like the *Fitbit Alta HR* are utilized as a heart rate monitor. The collected heart rates over time are then used to calculate the heart rate variability (HRV).

Heart Rate Variability (HRV)

HRV is the variation in the time of two consecutive heartbeats [46]. The Stila Android application uses HRV calculations of 10-minute windows to estimate whether a user is stressed or not [21]: A high HRV is generally considered to indicate a low stress level, while a low HRV can point to the user being stressed [50].

Computed Stress

A stress estimation which is computed from data delivered by non-invasive measurement of physiological stress symptoms is called “computed stress” [50]. In the case of the project Stila, computed stress is derived from HRV calculations. Computed stress can again be distinguished into computed eustress and computed distress.

Activity Tracking

A user can input her daily activities in the Android application to make a connection between her activities and the computed stress levels. Furthermore, the user can specify a self-assessment of her subjective feelings (positive or negative) during the activities. These include information about the perception of stress, the performance, and the overall well-being during the activities. The heart rate data, the activities and the self-assessments are then sent to the Stila back-end.

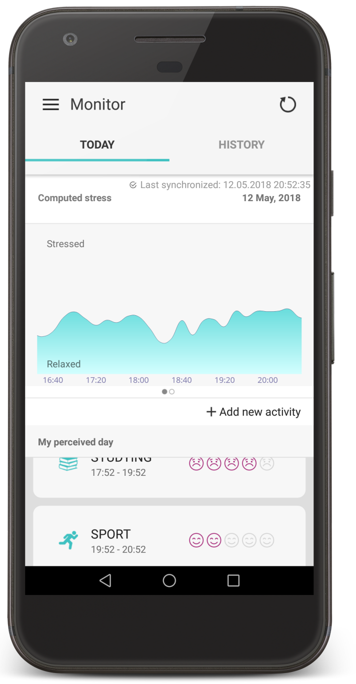

Figure 1.1: Stila running on an Android smartphone

Stress Classifier

In the back-end, the heart rate data and the self-assessments are the basis of a labeled data set, which is used to train a classifier. It is the goal of this classifier to distinguish between computed eustress and computed distress by matching heart rates metrics (such as the HRV) to the subjective feelings of the user. An appropriately trained classifier can decide whether the user was experiencing positive or negative stress without additional input of the user's feelings [17].

Outlook

The goal of Stila is thus to create awareness of the intensity and the different types of stress a user is experiencing for her to react to them accordingly. This awareness can not only help the user cope with her personal stress and prevent subjection to possible stressors, but could also be used on a greater scale in the university environment. Students could make their data instantly accessible to a lecturer, who could adapt the pace of the lecture to the stress levels of the students. If the lecturer recognizes peaks in the stress data, she could slow down and repeat the last concepts as they were probably not well understood by her class. A constantly high stress level among students of a course could also be an indication that the teaching in this course has to be improved or that material has to be cut from the curriculum. This and other scenarios require a convenient way to track the heart rates of users and the availability of enough labeled data to train the classifier.

1.2 Motivation

While the work of Patrik Hagen [17] has shown that it is possible to distinguish between computed eustress and computed distress automatically, it was evident that users did not track their daily activities often enough. A small number of tracked activities not only hinders the training of a eustress/distress classifier but also impedes a user from learning about the stressors in her daily life. This could consequently make it difficult for the user to cope with her stress adequately.

More user-labeled data could also enable the creation of classification models for each user. Currently, a single classification model is trained with data from all users. A personalized classifier is naturally expected to perform better than a single classifier for all users [17].

Furthermore, it would be advantageous to support a wide range of heart rate monitors, to simplify future studies or real-world applications of the Stila platform. More supported devices can be achieved by utilizing a widely used operating system for devices with heart rate sensors: This study concentrates on *Wear OS by Google* which supports smart watches by many different manufacturers and offers a well-equipped application programming interface which is based on Android.

1.3 Objective

The primary objective of this thesis is to increase **user engagement** by creating **behavior changes** in users of the Stila application: The Stila application should encourage its users to heighten their awareness of stress. An important point to consider is that the complete avoidance of stress is not a goal of the application. Stress, as previously described, can have positive as well as negative effects. A very low intensity of stress is thus not necessarily desirable [49].

The increased user engagement should lead to a more frequent tracking of the user's daily activities in the application. Consequently, the user's perception of her stress levels and the stress types should improve. A better perception enables her to develop coping strategies for negative stress and lead a healthier life by embracing eustress and controlling distress.

These goals are primarily reached by the use of a **smart watch application** as a companion to the mobile application in connection with approaches from **behavior change theory**. Figure 1.2 shows a conceptual outline of the objective and approach of this thesis.

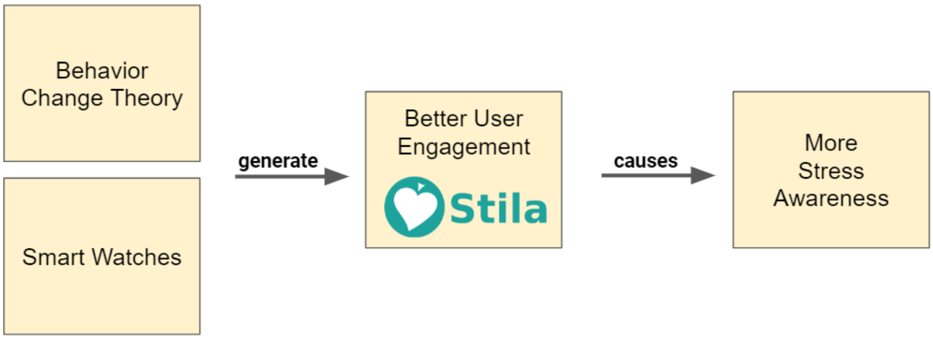

Figure 1.2: Conceptual outline of the objective and the approach

1.4 Thesis' Structure

The aforementioned objectives are achieved by first considering theoretical frameworks for user engagement and behavior change. Amongst these are:

- O'Brien's **Process of Engagement** [39]
- Fogg's **Behavior Model** [13]
- Fogg's **Functional Triad** [14]
- Design concepts for **Wearables** and **Smart Watches**

After the theoretical groundwork is explored, concepts are elaborated to improve the Stila application with the presented ideas. These include:

- **Stress feedback** mechanisms
- Approaches to **decrease physical and mental effort** while using the application
- **Smart notifications**

Subsequently, technical details concerning the implementation of the behavior-changing features are discussed in detail.

After that, it is reported on an evaluation of a prototype of the application. The evaluation is performed by conducting a user study: 43 users of the optimized application are compared against 43 users of a non-optimized version. Differences in the frequency of the activity tracking are examined, metrics about usage are evaluated, and questionnaires identifying the key reasons for improvements in engagement and stress awareness are analyzed.

This section is followed by conclusions about the measured data and the questionnaire before abstracting them to a more general level.

1.5 Related Work

Theoretical Models

As previously mentioned this work is based in part on Fogg's Behavior Model [13] and Fogg's Functional Triad [14]. These are theoretical frameworks to design persuasive technologies or behavior-changing software.

Behavior Change for a Healthier Life

There are several studies which focus on the use of engagement-increasing and behavior-changing software to positively impact the health of its users. An example of this is Consolvo et al. [7], who used software running on cellphones to encourage fitness behavior changes for a healthier lifestyle . Dennison et al. [9] pursued a similar goal and explored which features a smartphone application should have to support users in health-related behavior change. Furthermore, Lee et al. [28] developed a system which uses automated text messages based on Fogg's Behavior Model to increase the knowledge and receipt of cervical cancer screenings in populations with low screening rates. The avoidance of distress and the general awareness of stress which is the goal of this work could be described as an aspect of mental health or mental hygiene. To the author's best knowledge there are no studies which focus on behavior changes to improve health in these areas.

User Engagement and User Experience

There are multiple studies which explore the importance of a good user experience to engage users. One example of this is the work of Mandryk et al. [31] who investigated the role of user experience in entertainment technology.

Wearables and Smart Watches

The importance of user experience on smart wearables is also well researched. Examples include the work of Karahanoğlu et al. [26], who proved the relevance of personalization on smart wearables and a study by Lyons et al. [30] who investigated how smart watches are utilized by their owners. The use of smart watches with the explicit goal of positive behavior change has not yet been studied and therefore is a focal point of this work.

CHAPTER 2

Theoretical Framework

A solid understanding of theoretical concepts is essential in the search for new possibilities to change behavior and increase user engagement. In the following chapter ideas of several authors on user engagement, behavior change, persuasive technology and wearables are presented. Definitions are introduced to avoid uncertainties in the language and classifications are made to organize approaches. These are later used in the conception of the solution to decide which concrete measures can be taken to improve the Stila application in order to achieve the objective of this thesis.

2.1 User Engagement

2.1.1 Definition and Overview

Engagement is defined differently depending on the research field or application. Therefore it is necessary to specify the term in the scope of this work. In the English language *engagement* can be used to describe the initiation of contact or the concept of being occupied with something [42]. Zichermann et al. describes engagement as "[...] the period of time at which we have a great deal of connection with a person, place, thing, or idea" [62]. Quesenbery sees user engagement in software as a dimension of usability [44]. O'Brien establishes the term as "[...] a quality of user experiences with technology" [39].

In this thesis **user engagement** is defined as the ongoing commitment of the user to the application. This shows that engagement does not happen at a specific point in time but rather is a process.

2.1.2 O'Brien's Process of Engagement

O'Brien proposed a **Process of Engagement**, which users go through when being engaged by technology. The four steps of the process are described in the following:

1. **Point of Engagement**

   The point of engagement is the catalyst of the process. It can be triggered by "[...] aesthetic appeal or novel presentation of the interface, the users' motivations and interests, and users' ability and desire to be situated in the interaction and to perceive that there is sufficient time to use the application" [39].

2. **Period of Engagement**

   The period of engagement describes the time span a user is actively engaged with the technology. It is determined by the user's attention and interest. O'Brien proposes that these can be strengthened by the presentation of feedback and novel information on the interface. It is also beneficiary to let the user feel like she is in charge of the interaction.

3. **Disengagement**

   Disengagement happens when users make the " [...] internal decision to stop the activity, or when factors in the participants' external environment caused them to cease being engaged" [39]. Disengagement can be triggered by multiple circumstances, e.g. the lack of novelty, the absence of usability or external distractions. Disengagement can result in the cessation of the usage of the system or a continuing usage without engagement. It is also not purely negative. When a user fulfilled her intentions, she can disengage with a positive emotion because she feels successful and satisfied.

4. **Re-engagement**

   A user is re-engaged, when she returns to an application after previously having disengaged. A user also can be re-engaged if she abandoned the task before it was done. Re-engagement usually only happens if the user had a positive past experience with the application.

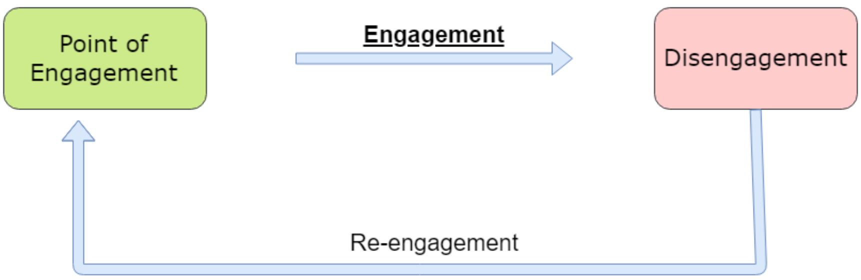

Figure 2.1: Simplified diagram of O'Brien's Process of Engagement

Figure 2.1 shows a graphical representation of the main ideas of O'Brien's process of engagement. O'Brien's model shows that engagement is a continuum which comprises steps that are characterized by attributes such as novelty, aesthetics, feedback and usability. O'Brien's process analyses engagement as a quality of user experience and provides a framework for understanding the mind of an engaged user. O'Brien does not advice system designers how to improve user engagement or how to identify why users do not perform a certain behavior. For that persuasive technology and behavior change theory can be used.

2.2 Behavior

2.2.1 Definition and Overview

Behavior

The term **behavior** is not easily defined because it is so widely used in the English language. Before examining the concept in the following chapters, behavior has to be specified in the context of this work:

Dretske defined behavior as performing a movement or bringing about an environmental outcome [11]. While this definition is certainly encompassing, it may be too abstract. Dretske offers another definition for **human behavior**: "Human behavior is what humans do" [11]. This explanation reads far more concrete and will be used in the following as a definition for the term behavior.

To examine what behavior means in the digital realm, one can look at **behavior informatics**. Behavior informatics is a scientific field which aims for an understanding of (human) behavior using computers. It encompasses the analysis of behavioral data as well as the building of behavior models from patterns in that data. It especially deals with behavior recorded using computational technology [5]. Within its scope behaviors refer to "[...] those activities that present as actions, operations or events as well as activity sequences conducted by entities within certain contexts and environments in either a virtual or physical organization" [5]. This definition is most fitting for this thesis, as it includes virtual environments.

Behavior Change

In this study **behavior change** is defined as the alteration of existing behavior or the uptake of new behaviors. The activity that should be executed after a behavior change is frequently referred to as the **target behavior** [40] [13].

Fogg et al. classified different types of behavior changes by placing them in one of 15 categories of **Fogg's Behavior Grid**. This categorization helps designers of behavior-changing products to analyze the target behavior with a systematic approach. After that, types of target behaviors can be matched with solutions. Fogg et al. divide behavior change into two dimensions: The *flavor* and the *duration* axis. The target behavior can be categorized into five different flavors each offering three possible durations. For easier reference, the flavors are named after colors [16]: Figure 2.2 shows Fogg's Behavior Grid with examples relating to healthy nutrition. The main benefit of the behavior grid is that after a target behavior is categorized, one can find pitfalls and challenges: A black behavior needs a very different approach than a green behavior. When designing for the black flavor, the target behavior has to be made as difficult as possible, while when developing a solution for the green flavor, it has to be thoroughly explained to make it easy. Furthermore, the strategies of causing a dot behavior differ widely to those causing a path behavior: The former needs a powerful one-time trigger, while the latter needs a complete shift in the identity of the target audience. This shows the importance of an accurate categorization of the target behavior.

	Green behavior Do new behavior, one that is unfamiliar	**Blue behavior** Do familiar behavior	**Purple behavior** Increase behavior intensity or duration	**Gray behavior** Decrease behavior intensity or duration	**Black behavior** Stop doing a behavior
Dot behavior is done one-time	Try eating dried seaweed for a snack today	Eat vegetables at dinner tonight	Increase mindfulness at lunch today	Eat only half of a hamburger tonight	Don't buy ice cream this time while shopping
Span behavior has specific duration, such as 40 days	Substitute quinoa for rice for one month	Drink water each morning this week	Eat more vegetables at dinner for two months	Eat fewer carbohydrates for one week	Don't use sugar in coffee for two weeks
Path behavior is done from now on, a permanent change	Lead a vegan lifestyle from now on	Take daily vitamins from now on	Increase healthy eating options in home	Decrease fried foods in diet from now on	Stop eating fast food forever

Figure 2.2: Fogg's Behavior Grid with examples from the field of healthy nutrition [16]

2.2.2 Fogg's Behavior Model

In 2009 B.J. Fogg [13] proposed a psychological model, which identifies three factors that control whether a behavior is performed or not. The goal of Fogg's Behavior Model (FBM) is to provide "[...] designers and researchers with a systematic way to think about the factors underlying behavior change" [13]. The FBM specifies the three dimensions of behavior as **motivation**, **ability** and **triggers**. The dimensions each have sub-elements which are explored in the following section. The model can help to understand why behavior change happens and identify which factors have to be altered to make it happen.

The main idea of the FBM is the following: To create a behavior change, **motivation**, **ability** and **triggers** have to be present and strong enough **at the same time**. Fogg describes the concept with the use of this pseudo equation [15]:

$$B = mat \tag{2.1}$$

In this equation B stands for behavior while m, a and t stand for motivation, ability and trigger respectively. The following paragraphs explain the three dimensions of Fogg's Behavior Model in more detail [13]:

Motivation

The first dimension of the FBM represents the motivation of the target audience. Simply put, the motivation reflects the willingness of a person to perform a behavior. To further structure this dimension, Fogg identifies three core motivators, each consisting of two sides.

- **Pleasure/Pain**
  Both pleasure and pain are primitive responses of the body and potent motivators. Products can increase pleasure in their users in order to motivate them. Fogg states

that, while pain is the logical opposite of pleasure it usually is not a good approach to increase motivation.

- **Hope/Fear**
  These motivators each represent an anticipation of something: Hope is the anticipation of something good, while fear is the anticipation of something bad. The connection to pleasure and pain is evident. Fogg believes that hope is the most ethical way to motivate someone.

- **Social Acceptance/Rejection**
  These motivators speak to the social instincts of humans. Things that cause social acceptance can motivate people. The avoidance of social rejection is equally as strong.

Ability

The second dimension of the FBM is the ability of a person to perform a target behavior. A higher ability means that the behavior is easier to perform. For this reason, the ability dimension is sometimes also called **simplicity**. Behavior (especially in the digital realm) can often be caused by making the required actions easier for the target audience. Fogg stresses that increasing the ability is not about teaching people new things. Behaviors that need excessive training often fail: Humans are "are fundamentally lazy" [13]. That is why it is often better to increase the ability of a person by making the target behavior less difficult. Fogg sees simplicity as a chain of six parts. If any of these elements fail, the behavior is no longer simple. The following briefly describes each of the elements in the chain of simplicity:

- **Time**
  If the target behavior requires more time than available, it is no longer simple.

- **Money**
  Behaviors that require large financial resources are harder to attain for people with little money to spend. The amount of money needed to break the chain of simplicity is different for each person: A wealthy person can spend more money before a behavior gets hard to do.

- **Physical Effort**
  A behavior that requires a lot of physical effort (e.g., body movement) may not be simple. When the behavior ceases to be simple is dependent on the person and the situation.

- **Brain Cycles**
  Target behaviors that require much thinking are usually not simple. This factor is again dependent on the person and the situation. Behaviors that are usually easy can become difficult when the mind is preoccupied, e.g., when driving a car.

- **Social Deviance**
  Behaviors are no longer simple, when they go against the social norm, break social rules or carry a stigma. The behavior "Do not wear clothes" will be very difficult for the target audience to perform because it is not socially acceptable.

- **Non-Routine**
  People find activities easier if they are done routinely. If the behavior is not performed often or is different every time, it can become more difficult.

As mentioned before, simplicity factors can differ depending on the person, the context and the situation: "Simplicity is a function of a person's scarcest resource at the moment a behavior is triggered" [13].

Triggers

The last dimension in the FBM are the **triggers**. This concept is also referred to as *cues*, *prompts* or *calls to action*. The purpose of a trigger is to inform the target audience that the time to perform the behavior is now. Triggers can be everything from a push-notification on a smartphone to a "Control Engine" light in a car. Some feelings like hunger or coldness can also be considered triggers for certain behaviors (e.g., eating and putting on clothes respectively). The way a trigger is presented to a person is not relevant, as long as it is associated with the target behavior. Fogg defines three kinds of triggers:

- **Spark**
  A spark is a trigger that is designed to cue the target behavior while simultaneously increasing the motivation of the user. An example for a spark-trigger is a notification which motivates by presenting a text message: "Buy now to save money!" would be a typical spark that motivates for the target behavior by creating hope. Sparks are typically used when the user probably has a certain amount of ability but might be currently unmotivated to perform the target behavior.
- **Facilitator**
  A facilitator informs the user that the target behavior is easy to do. It triggers the user while simultaneously making the behavior easier. A facilitator-trigger informs the user that the behavior can be done without needing much ability, like time or brain cycles. An example of a facilitator is again a certain kind of push notification: A click on the notification can take the user right into the application and to the screen where the target behavior can be performed. Facilitators should be used, when the user is motivated but has low ability.
- **Signal**
  Signals neither try to simplify the behavior nor to motivate the user. The purpose of a trigger is solely to remind. Triggers are especially suitable for behaviors which do not need a boost in motivation or ability. A traffic light is a common example of a signal trigger: It does not make the behaviors "Stop" or "Go" easier or manipulates motivation, it just informs the driver.

Fogg states, that especially when presented through mobile phones, people are most tolerant of signals and facilitators. Unsuccessful sparks can annoy users because they try to motivate for something that users do not want to do.

Overview

Figure 2.3 shows Fogg's behavior model graphically. The motivation dimension is presented on the y-axis and the ability dimension on the x-axis. To be analyzed, the target behavior can be inserted into the graph in a location according to the present motivation and ability. It is important to understand that this graphical representation of the FBM is only conceptual. Neither motivation nor ability can be measured objectively. The model's purpose is to help designers and researchers to think systematically about behaviors and not to measure each factor accurately.

After the occurrence of a trigger, the behavior is either executed or not. If the behavior was executed the motivation or the ability was high enough for the trigger to be successful. If the behavior was not executed either the motivation or the ability was not high enough. The line separating successful triggers from unsuccessful triggers is called the **action line** [15]. It is evident from looking at the graphical representation of the FBM that ability and

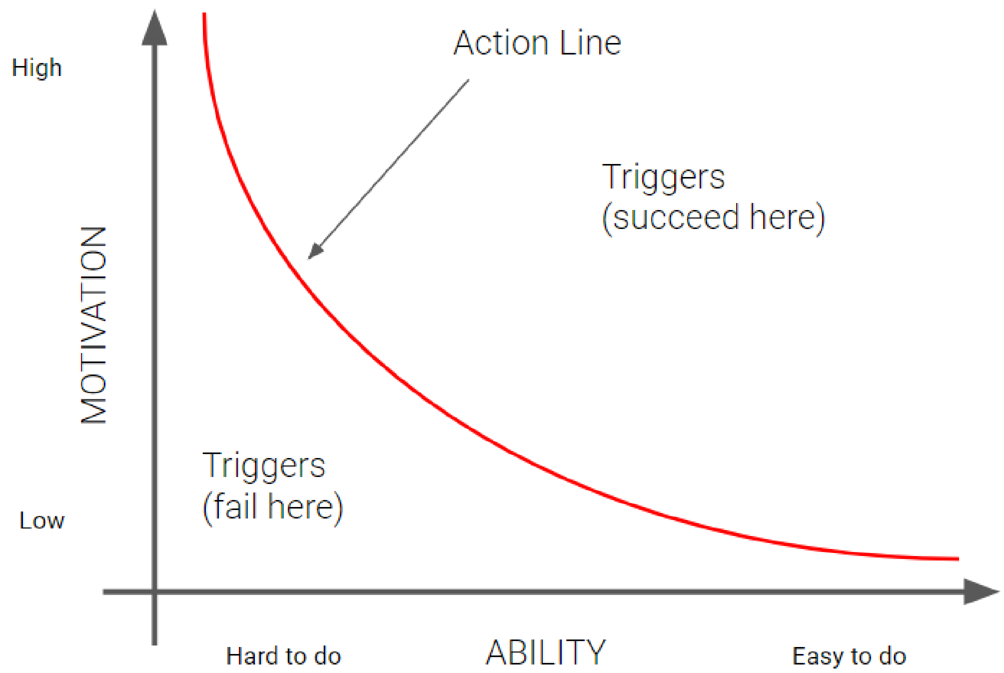

Figure 2.3: Fogg's Behavior Model (adapted from [15])

motivation can tradeoff: If the motivation is low, people might still perform the target behavior because the ability is high enough (i.e., simple to do) [13].

An example of this can be found in supermarkets: While waiting in the check-out line, customers are often exposed to further merchandise like sweets or chewing gum. The products in the vicinity of the check-out line are usually higher priced than in the back of the store. Most of the time shoppers have little motivation to buy these things because they already have everything they need. Nevertheless, consumers frequently buy these products because their ability to do so at this moment is extraordinary high. Shoppers do not have to move far and they do not have anything to do while waiting.

In the opposite case the motivation is very high and the ability is low. Again one factor can compensate for the other. An example of this is a mountain climber. Though the act of reaching the top of the mountain can be very cumbersome and difficult, her motivation to reach the top can compensate.

The shape of the action line shows that neither the motivation nor the ability should be zero or near zero for a trigger to be successful. There always has to be a minimum of motivation and the behavior must not be impossible to do. Another critical factor of the FBM is the timing of the trigger. To achieve execution of the target behavior the trigger has to be set at a moment when the motivation and the ability are high enough to move the behavior above the action line. Identifying these moments can be a challenging task for behavior-changing technology.

Concluding, it can be said that an architect of a behavior-changing product has the task of moving the target behavior above the action line and then to set an appropriate trigger. The higher the ability and motivation of a person are, the more likely she will execute the target behavior. Fogg states that often it is easier to increase the ability than to increase the motivation of the target audience: “People often resist attempts at motivation, but we humans naturally love simplicity” [13].

2.3 Persuasive Technology

2.3.1 Definition and Overview

Persuasion

Before the term **persuasive technology** can be explored, **persuasion** in this context has to be defined. Fogg describes persuasion “[...] as the attempt to change attitudes or behaviors or both” [14]. The term *behavior* was defined in the last chapter, which only leaves the term *attitude* for explanation. **Attitudes** can be defined “[...] as general evaluations people hold in regard to themselves, other people, objects, and issues” [43]. The term *persuasion* is differentiated from the term *conviction* by looking at the strategy that is used. Persuasion relies more on the emotions of a person and often utilizes more symbolic factors, while conviction is achieved using logical reasoning to change attitudes or behaviors of a person [36]. Harjumaa et al. distinguish different kinds of persuasion dependent on who fills the role of persuader and persuadee [19]:

- **Interpersonal Persuasion:** The persuasion involves two or more people who directly interact with each other.
- **Computer-mediated Persuasion:** The persuasion is performed by people but communicated with the help of computers (email, websites, etc.).
- **Human-Computer Persuasion:** The persuadee is persuaded by the use of computer technology.

Human-computer persuasion differs from the other two in one point: It is not always clear what the identity of the persuader is. If a persuasion attempt is successful, it can have three different outcomes [36]: The reinforcement, the changing or shaping of new behaviors and attitudes.

Persuasive Technology and Captology

Building on the definition of persuasion, **persuasive technology** is defined as “[...] any interactive computing system designed to change people’s attitudes or behaviors” by Fogg [14]. Oinas-Kukkonen et al. define persuasive technology differently by narrowing it down and placing it against the backdrop of ethics: For these authors, persuasive technology is “[...] a computerized software or information system designed to reinforce, change or shape attitudes or behaviours or both without using coercion or deception” [41].

The main goal of the scientific research concerning persuasive technology is to learn how behavior change can be automated [13]. Fogg also coined the term **captology**, which contains an acronym of the sentence “computers as persuasive technologies” [14]. This definition places the focus on computers as the technology facilitating the persuasion. Fogg suggests that captology only uses human-computer persuasion. Computer-mediated persuasion does not fall into this category because the technology used is not specifically designed to persuade (such as emails). To be considered as captology, the technology also has to have a planned persuasiveness which does not stem from a side effect [14]. In the scope of this thesis the term *persuasive technology* and *captology* are used interchangeably, as outside of Fogg's work the term captology is rarely used.

Advantages

When comparing persuasive technology with purely interpersonal persuasion, several advantages that encourage the utilization of computers can be found [14]:

1. **Persistence:** In many areas of application, computers can be far more persistent than humans when completing a task. This also applies to persuasion.

2. **Anonymity:** Computers can persuade someone without having to know their identity. This can be an advantage in persuasion concerning sensitive topics like drug addiction among others.

3. **Data Volume:** Computers can work with and analyze huge databases. This ability can enable them to make more persuasive suggestions and arguments than humans.

4. **Modalities of Influence:** Information meant to persuade can take many forms. Technology can present information in far more modalities than humans. While humans are usually limited to persuade by speech or text, technology can utilize video, audio, simulations and many more media types.

5. **Scaling:** Humans can usually only persuade a few people at a time. Technology, on the other hand, can rapidly scale to reach millions of people.

Another advantage of persuasive technology stems from the advent of **ubiquitous computing**. The term was coined by Mark Weiser and describes the concept of enhancing "[...] computer use by making many computers available throughout the physical environment" [60]. Fogg argues that ubiquitous computing improves the persuasive powers of technology because these computers can reach the persuadee everywhere and always. Ubiquitous computers can even go where human persuaders are not welcome (e.g., private spaces) or physically cannot go [14].

Furthermore, persuasive technology has an important advantage over persuasion via traditional media: Through interactive persuasion techniques computers adjust to the user's needs or situation. For instance, persuasion approaches in the field of drug addiction can be tailored to the severity of the addiction. A light persuasion attempt might not work on a very addicted person, while an overly strong persuasion attempt could deter a less addicted person. Traditional media, like text, video or audio cannot react on its user in the same way. This makes traditional approaches less effective [14].

Applications

Behavior-changing technology can take many forms and can be applied in various areas. The most popular application of persuasive technology is in the **health** field. Concrete examples range from helping people to quit smoking or eat healthier to specialized solutions for certain ailments [18]. Closely related to the health field is **personal fitness**. Persuasive technology can help people achieve their goals by keeping motivation high or making training more accessible. Examples include applications that encourage their users to move more [7] and fitness trackers that seek to motivate their owners to achieve their goals [34]. Persuasive technologies can also help in **environmental protection**. It has been successfully used to encourage sustainable behavior in everyday life [35]. Many features on **e-commerce** websites also have a persuasive purpose (suggestion systems, 1-click buying) and can, therefore, be counted as persuasive technology [14]. Even a car's acoustic signaling when the driver is not buckled up is a simple example of persuasive technology in the field of **safety** [54].

The multitude of applications areas shows the relevance of persuasive technology. With the continuing rise of ubiquitous and mobile computing, persuasive technology can be expected to enter even more spaces in the near future.

2.3.2 Fogg's Functional Triad

After covering the definition and purpose of persuasive technology, this section answers the question *how* technology can be persuasive. To address this question, Fogg developed the **Functional Triad** [14]: The functional triad "[...] is a conceptual framework that illustrates the different roles that computing technology can play" [14]. Fogg defines three roles that computers can take: **Tool**, **media**, and **social actor**. Building on this categorization, computing technology can be analyzed regarding their persuasiveness.

The functional triad lets designers of persuasive technologies categorize the features of their products depending on whether they persuade as a tool, as a medium or as a social actor. The same persuasive technology product can fill different roles of the functional triad at the same time. Fogg suggests persuasion techniques for each element in the functional triad. After a designer has categorized her persuasive technology in the functional triad, she can use these suggestions to discover flaws and improve the persuasive power.

Figure 2.4 shows an overview of the functional triad and its components. The following section highlights each of the corners in detail.

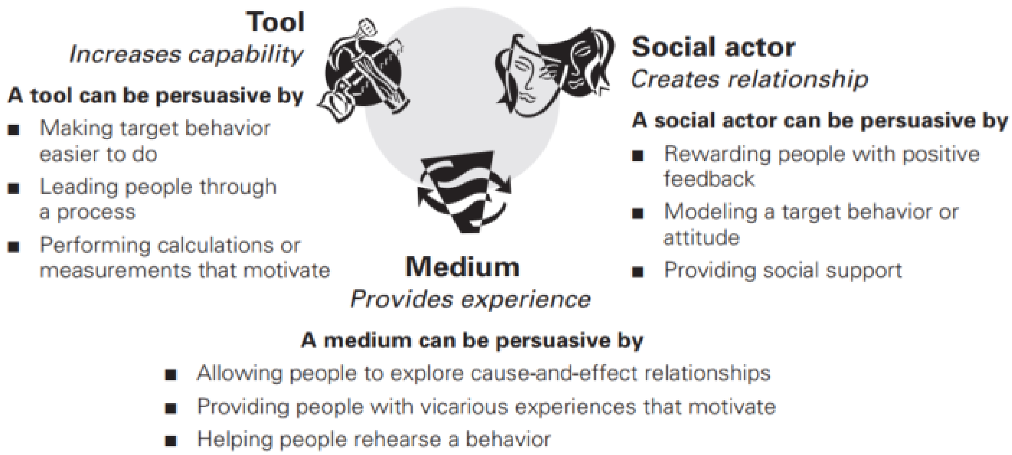

Figure 2.4: Overview of Fogg's Functional Triad [14]

Computers as Social Actors

Computers can act socially when humans see them as a living being. This ranges from humans screaming at computers when they are not working to literally feeding and caring for computers in the form of Tamagotchis.[1] Humans are prone to react to social cues, so persuasive technology can use these to generate social responses. For a technology to be persuasive as a social actor, it has to create a relationship with its user. Fogg identifies five kinds of social cues that can be used in persuasive technology [14]:

1. **Physical Cues:** Physically attractive technologies can potentially be persuasive. This factor refers to the device itself as well as to its interface. Responsible is the so-called *halo effect*, which lets users experience an attractive product as capable, intelligent and reliable [20].

2. **Psychological Cues:** These cues can make humans believe that a computer has emotions, motivations, and a personality. Designers can use this by building technology that is in some way human-like. An example for this is using empathic language in user interfaces ("I am very sorry, but the App crashed") or utilizing representations of the human face to convey information. If applied correctly, these techniques can profit from the **principle of similarity**. This principle states that people are easier persuaded by people (or in this case computers) similar to them [57].

3. **Language:** Computers can use language to act socially. This involves written and spoken language. Written language is often found in dialog boxes in the interface, e.g., when it asks the user a question ("Are you sure you want to delete this file?"). Spoken language can be facilitated by using text-to-speech software or pre-recorded audio files. Developers sometimes give their product a unique personality by deciding how it "talks" to the user. Fogg concludes that a powerful, persuasive usage of language is **praise**. As mentioned earlier, many fitness trackers use praising language to persuade users to move more.

4. **Social Dynamics:** Computers often act in accordance with certain social dynamics. An example of this is a personal greeting on a website. While this does not fulfill any technical purpose, it may give the user a feeling of a social relationship with the website.

5. **Social Roles:** Technology assuming roles of high social reputation can have higher persuasive power. This power stems from the human tendency to respect authority and expect leadership from certain professions and roles. This phenomenon is frequently used by including words like "doctor" into names of anti-virus software.

Computers as Media

Fogg states that computers can be used as **symbolic** and **sensory** media. Both of these usages can help to facilitate behavior change. Symbolic media uses symbols to convey information (charts, symbols, text, etc.). Sensory media provides sensory information to the human body (video, audio, virtual reality). The way to successfully persuade lies in providing an experience. Computers can be used to simulate complex situations which enables their users to explore cause-and-effect relationships. The graspable experience of these relationships can enable users to understand arguments of the persuader better.

[1] Tamagotchis are small computers housing digital pets, which were popular in the late 1990s.

Computers as Tools

The most important element of the triad for this work is the use of computers as tools. A computer used as a tool can increase the capability of its user by making things easier to do or creating possibilities that would not be viable without technology. Tools can be used to enable a user to perform the target behavior easier. Fogg defines a persuasive technology tool as "[...] an interactive product designed to change attitudes or behaviors or both by making the desired outcome easier to achieve" [14]. Tools can make an activity easier and more efficient by reducing the time a user needs to fulfill a task. Fogg subdivides persuasive technology tools into seven types. A persuasive technology product can use several tool types at once to achieve its goal. The persuasive tool types are explained in detail in the following [14]:

1. **Reduction:** Reduction technology persuades by simplifying. It reduces complex behaviors to simple behaviors. Reduction increases the benefit/cost ratio for the user by decreasing the cost. It can also improve the user's self-efficacy and even her belief in her ability to perform the target behavior. Fogg concludes that these factors lead to a positive attitude about the target behavior which results in the user trying harder to adopt the behavior and perform it more frequently. A popular example of persuasion through reduction tools is online retailer Amazon's *1-click buy* feature: A customer can buy a product by clicking only once. She does not have to go through a lengthy checkout to order the item. This reduction is meant to persuade customers to buy more products from Amazon, as it is easier to use than other web shops.

2. **Tunneling:** The act of guiding users through a process or experience is called *tunneling*. A tunnel is a predetermined sequence of actions or events that the user passes through. By entering a tunnel, the user gives up a certain level of self-determination: She is expected to complete the tunnel before using the technology. A tunnel constitutes an excellent opportunity to persuade. Because the designer of the tunnel controls the experiences and events, users become captive audiences. This effect can make users more easily persuaded. As a positive side effect of tunneling, it can be made easier for users to go through the experience because they are guided by the tunnel and have less chance to get lost along the way. An example of tunneling in software is an installation dialog found in desktop programs. The user is guided through the progress by clicking "Next" buttons and providing necessary information. The tunnel makes it easier for the user to use the program. On the other hand, the tunnel could be used by the designer to present the features of the program or ask for the user's data in an optional registration dialog.

3. **Tailoring:** Tailoring describes persuasion through customization. Persuasive technology which uses tailoring provides different information to different users. Information can be selected depending on the user's needs, interests, personality or usage context. Tailored or customized information is more effective in persuasion and behavior change because the user pays more attention to it. Furthermore, the user can quickly find information that is relevant to her. An example can again be found in online commerce: Many web shops tailor user experiences by showing customers products they might be interested in. These recommendations can be made by using data from previous purchases.

4. **Suggestion:** Fogg describes suggestion technology as "an interactive computing product that suggests a behavior at the most opportune moment" [14]. Persuasive technology can cue a behavior by pointing out a good time for it. How to identify these moments is as difficult as it is essential. Scientific research summarized by Fogg suggests that several factors determine the right point in time. These include the mood of

the user. If a user is in a good mood, she is more likely to perform the target behavior. Excellent results can also be achieved when a person feels indebted to the technology because a favor was given to them by it or they already denied recent requests. It is crucial to consider elements of the environment when picking a moment to persuade. Some of these elements like the user location or the current time can be found out by modern technology. Other factors (e.g., whether the user has time to spare) are far more difficult to be analyzed by technology. Examples of suggestion systems include computers that ask the user to install an update when it concluded that she is not doing important work. Another well-known application is speed monitors on streets that show motorists their current velocity while suggesting to slow down if they go too fast. Here, the suggestion to keep under the speed limit is made at an opportune point in time: While driving. The connections from suggestion technology to the concept of **triggers** in Fogg's Behavior Model is apparent.

5. **Self-Monitoring:** The potential for behavior change can also be increased by self-monitoring capabilities. People can use technology to monitor themselves, their progress and performance relating to a goal. Fogg suggests that self-monitoring can help in this by eliminating the need to track performances or other variables. Self-monitoring shows users how far along they are in performing the target behavior and can help them continue until they achieve it. Fogg recommends that self-monitoring technologies ideally work in real time and give its users an ongoing stream of data about their state. Furthermore, self-monitoring can help users to learn about themselves. Many people desire such information, which can make the usage of a persuasive technology intrinsically motivated. A prominent example of a persuasive technology of the self-monitoring type are fitness trackers. Many of these devices allow the monitoring of daily step counts and calorie consumptions based on the heart rate. This information can be used to persuade users to be more active in order to achieve their goals.

6. **Surveillance:** The counterpart to self-monitoring is surveillance. While self-monitoring is performed by the user herself, surveillance is used to monitor her actions by third parties. Fogg defines surveillance in the context of persuasive technology as "[...] any computing technology that allows one party to monitor the behavior of another to modify behavior in a specific way" [14]. Observation seems to have a substantial effect on the behavior of people. It is not uncommon to behave differently when one is being watched. Surveillance can only unfold its persuasive power if it is not hidden but overt. An example for the persuasive power of surveillance methods includes high score lists in games: Many gamers might be persuaded to play a game more often and try harder to reach the top of a high score list, which is visible for all other players.

7. **Conditioning:** Conditioning in the context of persuasive technologies describes the act of " positive reinforcement to shape complex behavior or transform existing behavior into habits" [14]. Fogg explicitly excludes punishment as a form of conditioning on the grounds of it being unethical. Positive reinforcement in the form of a reward should occur right after the target behavior has been performed to be most effective. Not every performance of the target behavior has to be rewarded. An irregular reward can even make the persuasive power of the system stronger. An example of this are slot machines: They reward the target behavior (putting money in the machine) by irregularly paying out a large jackpot. This form of persuasion is so strong that people can even get addicted to the target behavior.

Connecting Fogg's Behavior Model to Fogg's Functional Triad

It makes sense to compare Fogg's Behavior Model to Fogg's Functional Triad, as the models seem to have some overlap. The main difference between the two is the purpose: Fogg's Behavior Model illustrates how one can think about behavior and how the human psychology causes behavior change. It helps researches and designers to " [...] think more clearly about behavior"[13].

Fogg's Functional Triad, on the other hand, describes what roles computers can play when trying to change behavior as persuasive technology [14]. Fogg's Behavior Model answers the question *what* behavior change is. The functional triad shows *how* behavior can be changed using technology.

The **tool** corner of the triad hold the means to increase the **ability** of a user to execute a behavior. The **social actor** and the **medium** corner on the other hand offer possibilities which promise to increase the **motivation**. The **trigger** dimension of Fogg's Behavior Model is not included in the triad. The closest factor resembling triggers are **suggestion technologies** in the **tool** corner of the triad.

2.3.3 Ethics

Persuasion as a term can be located between manipulating and convincing someone [54]. This definition shows how negatively connoted persuasive technology can be. However, persuasive technology itself is not necessarily unethical. It depends on the purpose and the intended use. Most of the time persuasive technology faces the same ethical issues as persuasion attempts without technology. Relevant for the ethical evaluation of a such persuasion attempts are the **intentions** and the **methods** [14]. Whether an intention is perceived ethical can differ from culture to culture and from person to person. Some methods are recognizable as unethical. Examples of this are extortions or other methods of forcing a behavior change. Another critical factor to consider is the target audience of the persuasion attempt. Ethical persuasion should not target children, elderly or other groups that may not be able to identify a persuasion attempt [14]. It is essential to keep a positive outcome for the user in mind while designing persuasive technology. It should also be remembered that the user is entitled to her own free will and persuasive technology should not hinder her in this respect. It is arguable, whether it is ethical to stop somebody from e.g. smoking, even if the person does not want to stop. To avoid such complicated cases, it seems advisable to design persuasive technology that allows the user to decide whether she wants to use it or not. A good example of this can be found in the usage of **tunneling**. A creator of ethical persuasive technology should always let the user decide whether she wants to continue in the tunnel and clearly show her an exit if she wants to stop [14].

Furthermore, Spahn developed a small ethical framework. It contains four guidelines which shall be followed when designing persuasive technologies [54]:

1. **Comprehensibility**
   Any feedback given by the system should be presented in such a way, that the user can understand it.
2. **Truth**
   Persuasive technology should only give true information.
3. **Truthfulness**
   Feedback mechanisms should be as reliable and accurate as possible.

4. **Appropriateness**
   The method of reaching a persuasion goal should be appropriate to this goal.

2.4 Wearables

2.4.1 Definition and History

Wearable Technologies, **Wearable Computers** or for short **Wearables** have a long history. Starting from abacuses worn on a finger in 17th century China [56] over smart watches to smart glasses that augment reality in the 21st century. The advent of wearable computers started in the 1980s with the availability of small and power-efficient microchips [32]. This progress moved computers from immobile to portable and then to wearable [33].

Starner defines wearable computing as follows: "Wearable computers are physically close to the user, highly portable, quickly accessed, and designed to consume a fraction of the user's full attention" [55]. Wearable computing pioneer Steve Mann described three criteria that define wearable technology [32]:

1. **Eudaemonic Criterion:** A wearable computer is worn, not carried. Others can regard it as part of the wearer. It is not tethered to a static power source and highly portable.

2. **Existential Criterion:** The computer can be controlled by the user. The control does not necessarily need conscious thought or effort.

3. **Ephemeral Criterion:** The device operates in real-time and is always active. The user can interact with it at any time.

Personal Digital Assistants (PDAs) and therefore smartphones are per definition not wearable devices. Although a smartphone can be worn close to the body in a pocket (eudaemonic criterion) and is always active (ephemeral criterion), it takes a rather high level of effort to remove it from the pocket and to use it (existential criterion). It is difficult to use a smartphone while doing other tasks, like working or walking. It is therefore not a wearable device [55]. Wearable computers are ever-present. They allow a much higher level of synchronization with available information than traditional computers [33]. Starner states that wearable computing devices aim to be an "intelligent assistant that augments memory, intellect creativity, communication, and physical senses and abilities" [55].

Contemporary Examples

Examples of modern wearable technology include fitness trackers, smart watches, virtual and augmented reality headsets as well as clothing made from smart fabrics. Figure 2.5 shows several contemporary representatives of wearables: On the left, the *Google Glass Enterprise Edition*, which is an augmented reality headset aimed at businesses can be seen. In the middle the *Commuter X Jacquard* smart jacket by Google and Levi's is depicted. It allows its users to control several functions of their smartphones by touching the sleeve of the jacket which is made of conductive yarn. On the right, the *Moto 360* smart watch by Motorola is shown.

Adoption

The adoption of wearables is often slower than that of other emerging technologies. An example of this is the massive backlash Google faced when it released the first developer preview of the Google Glass Explorer Edition. Many people did not like the fact that the

Figure 2.5: Contemporary wearable technologies: Google's Glass EE [23], Levi's Commuter x Jacquard [29] and Motorola's Moto 360 [38].

device could record video without a visible indicator [56]. One explanation for the difficult adoption is that wearables penetrate the personal space of people. Per definition, wearable technology is worn and active almost all the time. It is a bigger decision to adopt a device which is worn all the time than one that is only used from time to time.

Furthermore, Bodine stresses that the acceptance of wearable devices relies on the functionality and the design. Especially the functionality and the benefits have to be made apparent to potential users to achieve widespread adoption of a wearable technology [3].

2.4.2 Smart Watches

As already mentioned, many modern wearable computers can be found in the form of smart watches. This section examines how a smart watch can be defined and what advantages it has.

Definition

Xu defines **smart watches** by separately examining both of the words. The term *watch* stems from the fact that these wearable computers are " [...] designed to be worn on the wrist and provide quick access to the time" [61]. Whether a watch is *smart* or not is dependent on the functionality of the device. On the low end of the functionality spectrum, there are analog watches, which usually only show the time. Digital watches offer more functionality by providing features like a stopwatch or an alarm. On the high end of the functionality spectrum, modern smart watches with fully capable computing hardware can be found [61]. It is not fully clear when a watch starts to be *smart*. Is an analog watch that can measure the steps and heart rate of a user a smart watch? Lyons argues that for a watch to be smart it has to have the ability to install and run applications. Consequently, it is not locked into a fixed set of functions like, e.g., a traditional digital watch [30]. Xu explicitly excludes activity monitors (fitness trackers) from the smart watch category. These devices do not allow the user to increase the functionality of the device by installing further software [61]. Xu states that the connection to a smartphone can provide the smart components of a watch. The pure processing power of the watch is thus not a differentiator. Depending on the application the watch can act as a thin client and rely on the processor of a tethered smartphone [61]. In general, smart watches are positioned as a companion to smartphones. The watch can be paired via Bluetooth and used to, e.g., display notifications from the phone [30].

Within the scope of this thesis smart watches are defined as a wearable computer that resembles a watch which can install and run software that expands its original functionality and works best when tethered to a smartphone. The last part of the definition states that the watch does not have to be connected to a smart phone at all times, but a connection can further increases the functionality of the watch.

Examples of watches that are covered by this definition include the aforementioned Moto 360 by Motorola, the Apple Watch, and the Fitbit Versa. Each of these smart watches come with a pre-installed operating system which supports third-party applications. The operating systems used by the watches are Wear OS by Google, WatchOS by Apple and Fitbit OS respectively.

Advantages

Multiple factors speak for the usage of smart watches as wearable computers. One reason for this is that people are already used to wearing a watch on their wrists [45]. Also, smart watches do not look as unusual as, e.g., an augmented reality headset, which further eases their adoption by a broad audience. Furthermore, smart watches allow their users to view information with a simple flick of a wrist [45].

2.4.3 Interaction

This section examines the interaction with wearables. The most important points are extracted from the relevant literature. The focus lies on the interaction with smart watches. Nevertheless, most of the below-mentioned factors are applicable to all wearables that possess a small touchscreen. In the following, the topic is split into three categories which contain summarized interaction-design considerations for wearables by different authors.

General User Experience

1. **Expect Less Attention:** On highly mobile wearable devices the user experience is more personal than on any other computing product [26]. This effect is caused by the fact that the device is worn on the body and carried everywhere the user goes. Consequently, a traditional WIMP[2] interface might not be the best solution. These interfaces require a high amount of attention and have to be in the primary focus of its user. While this is certainly the case when operating a desktop computer, this can not be expected for the usage of wearable computers. Starner states that interfaces of wearables should "provide the most support for the smallest investment of attention diverted from the user's primary task" [55]. In this case, the primary task is the user's current activity, e.g., walking down the street. Because the wearable is always around, it must not demand much attention.

2. **Expect Interruptions:** Because wearables are frequently used without the user's full attention, designers of wearable technologies have to expect that users will be interrupted while using them. When a wearable is used while walking down the street,

[2]**W**indows, **I**cons, **M**enus, **P**ointer

its user will frequently look up to see where she is going. Users can also be distracted by their environment. The environment is a much more important factor than with traditional computers because it changes rapidly when using a wearable device. Designers of wearable technology can counteract these problems by simplifying the user interface and providing automatic save functions that prevent the user from having to start again after an interruption [53].

3. **Limit Functions:** On a wearable device, the number of functions should be limited to essential ones. Features that can not be described as essential should be relegated to a companion application on a mobile or desktop device [53]. While it may seem counter-intuitive to reduce the number of functions an application offers, it is important to consider that too much content makes an interface less clear. Especially on a very small display, a cluttered interface can have negative effects on the user experience and the ability of the user to find features. Because most wearables are tethered to a smartphone, it makes sense to migrate less important features to its larger screen.

4. **Strive for High Learnability:** Wearables have less (or no) display space to show tutorials, help-texts, and documentation. Therefore it is important to make software for wearables easy to understand and fast to learn. If these products cannot be learned in a few minutes, they even face the risk of being abandoned [2].

Input

1. **Avoid Fat-Finger Problems:** The issue of on-screen objects being smaller than the user's finger (commonly known as the "fat-finger problem") is also present on wearables [6]. Therefore it is important to design interfaces with large clickable elements. There should also be enough spacing between multiple buttons so that the user does not accidentally click the wrong button [48].

2. **Optimize for One-Handed Use:** Especially when creating interfaces for smart watches, it is important for the designer to consider that the device will be used with only one hand. A smart watch is usually worn on the wrist, what makes it impossible to use both hands to interact with the watch. Input modalities known from the smartphone like two-handed typing are not possible on a smart watch [45].

3. **Avoid Text Input and Complex Tasks:** Due to the small form factor of most wearables, it is advisable to avoid text input and complex tasks [53]. Complex tasks should be carried out on a device that has the full attention of the user like a desktop computer. If a text has to be created, it is best to write it on another device (e.g., a smartphone) and then send it to the wearable [45].

Navigation

1. **Optimize for Easy Navigation:** On small screens, it is important to remember that the interface becomes temporal. Only a part of the interface can be shown at a time, which creates the need for navigation. The interaction design must support an easy switching between screens and levels of hierarchy as well as scrolling [53].

2. **Allow Quick Returns to the Watch Face:** When designing software for smart watches, the user should always be able to return to the watch face quickly. Showing the time is the primary task of a watch and should always be focused upon [45].

3. **Allow Easy Returns to Previous Screens:** The user should always be able to return to the previous screen via a history stack. This behavior is commonly known from web browsers and is thus also expected in other interfaces [45].

4. **Use Gestures:** Because of the Fat-Finger problem and the small display size, it is advisable to use gestures when designing navigational elements. Hyperlinks take up space and are difficult to select. Possible gestures include swiping motions and wrist flicks in the case of smart watches [45].

5. **Simplify Common Tasks:** The most common tasks should be the easiest to launch for the user. This will enable her to use the product efficiently and without the need for much focused attention [53].

2.4.4 Design and Aesthetics

Karahanoglu describes the importance of design and aesthetics in wearables as follows: "For a smart wearable, hedonic qualities are as important as pragmatic qualities" [26]. Visually pleasing interface design can create a positive first impression, which in turn increases the likelihood that the device or software will be used. Wearable computers and their interfaces can be a personal statement of their users [52]. The design of a wearable can reflect the user's taste [55]. In the case of smart watches, this can be observed by the availability of many watch faces for different devices. A user can not only express her personality when buying a device but also change the watch face depending on her outfit or activity. A businesswoman might use a classic analog watch face in the office and change to a modern digital watch face when going hiking on the weekend.

It is also known that visual aesthetics of interfaces on computers can increase user satisfaction and pleasure [53]. Satisfaction and pleasure are especially important on wearable devices because these tend to be more personal than desktop computers. The user might look at the interface many times per day in different situations. This makes it important for the wearable to present a pleasing interface so that the user will not abandon it.

It is challenging to accurately measure the aesthetics of an interface or optimize the user's satisfaction with a design. These factors are inherently subjective. There is, however, design advice identified by Shneiderman to improve the aesthetics of colors, texts, and data visualization [53]:

1. **Use Color Conservatively:** Too many colors can be overwhelming for the user. The number of colors should be limited to four per display and seven in the whole application.

2. **Be Consistent with Colors:** Use the same colors for the same actions. If a "Cancel" Button is red on one screen, it should be red on all screens of the application.

3. **Be Consistent with Positioning:** If an element (e.g., a button opening a menu) is present on multiple screens, it should be at the same location in the display.

4. **Use Guideline Documents:** Many operating systems and organizations offer guideline documents for designers. The documents frequently include advice on what colors should be used and how an interface should look. The usage of these can help to offer the user a coherent design experience across multiple applications, which improves the user experience. An example of such a document is Google's *Material Design* guideline which can be used for designing applications on different devices.

5. **Use Concise Writing:** Especially on small displays, it is important to limit the number of words and letters on each screen. This limitation will make the display look tidy and easier to use.

6. **Use the Visual-Information-Seeking Mantra:** Shneiderman's Visual-Information-Seeking Mantra describes how interaction with information visualizations should be implemented: "Overview first, zoom and filter, then details on demand" [53]. In other words, a zoomed-out version of the graph should be presented to the user at first. After that, the user should have the possibility to zoom into the visualization and filter data (if applicable). Further details about the visualization can then be revealed on demand, e.g., by the use of a context menu.

2.4.5 Cognitive Wearables

Sullivan defines **cognitive wearables** as follows: "Cognitive wearables are devices that either measure or affect our cognition" [56]. These types of wearables are the counterparts of fitness trackers for the mind: Instead of measuring bodily performance they measure emotions and feelings. Like fitness trackers, cognitive wearables measure biological indicators to gather data.

Examples

Cognitive wearables are an emerging field, but there already exist multiple examples, some of which are even commercially available: The *muse* headband uses electroencephalography (EEG) to measure electrical signals in the brain of the wearer. Muse helps the user to meditate better. The headband works together with a smartphone application that gives the user audible feedback on how deep her meditation is. When the muse system measures strong brain waves, the wearer hears a loud wind that gets softer and quieter the more the user relaxes [56]. Figure 2.6 shows the muse headband with its iOS application.

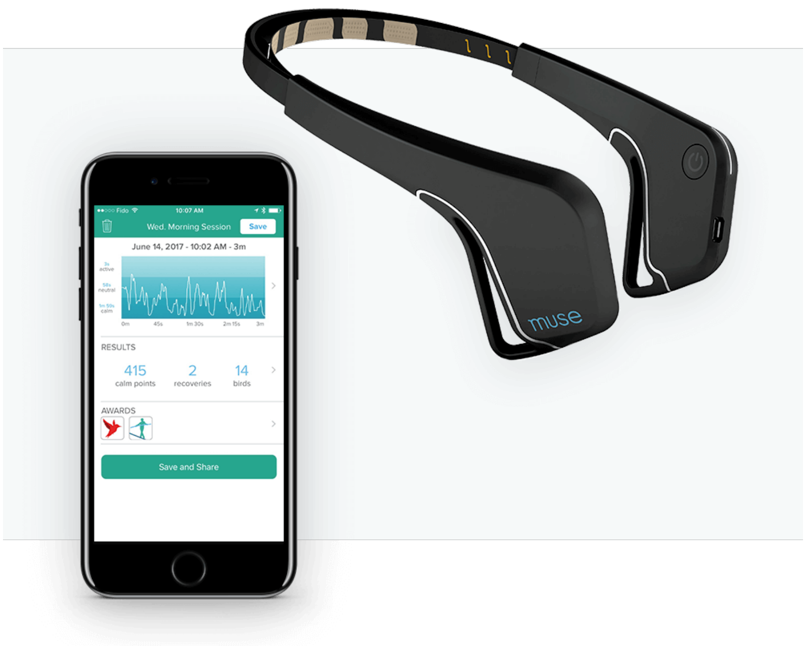

Figure 2.6: The muse headband with its iOS application [22]

Another example is the *Spire Stone*, a small device which can be clipped on the user's clothes. The device can measure breathing patterns and thus identify emotions like anxiety. Spire comes with a smartphone application, which informs the user about their mental status (tense, calm, focused, active) and gives recommendations on how to positively affect the current status. Both wearables mentioned above have a disadvantage: They have to be bought for their specific use case. They do not allow flexibility in how they are used. Both devices are also considerably expensive, what (in conjunction with their limited use cases) makes them unattractive for many potential buyers.

A scientific example of a cognitive wearable was developed by Webster et al. [59]: Here a traditional wearable was used to measure cognitive data. Webster et al. used a *Hexoskin* exercise shirt with 34 different sensors to predict changes in mood, motivation and behavioral context of the wearer in a smartphone application. The sensors measured movements of the body, the heart rate and respiration and other biological indicators. This example shows that a wearable does not have to be built cognitive but can also be made cognitive by the usage of software.

Challenges

Cognitive Wearables face some special challenges that other types of wearables do not: The first problem lies within the **language** that is used. Fitness trackers use well-known measurement units to represent data. Users can find out how many steps they have taken, how many calories they have burned and so on. Cognitive wearables do not have a fixed language. There are no units to measure how stressed someone is or how deep their meditation is. How designers of cognitive wearables solve this problem is usually dependent on use case at hand [56].

The second difficulty stems from the **granularity of cognitive information**. A wearable that measures focus and concentration deals with information on a different level than a wearable that measures burned calories. For the user it is often not transparent why she was not concentrated at a certain point in time, while it is certainly easier to understand why more calories were burned on a day with much physical activity. The way many wearables solve this problem is by letting the user track her circumstances and events to allow her to make connections between measured data and her situation at the time [56].

2.4.6 Persuasive Wearables

Wearables are uniquely well suited to act as persuasive tools because they are taken everywhere the user goes. Fogg states that mobile devices "[...] are not adopted, but married" [14]. Because of their mobility, they can try to persuade the user at the right moment while offering higher convenience in the interaction [14]: Many wearables have access to location and activity sensors which makes them able to start persuasion attempts at a time and place when the user is highly susceptible to them. The higher convenience stems from the fact that they can be used almost everywhere. Wearables are even able to fill "mental whitespace" in moments of downtime [14]. E.g., when a user is waiting for her train to arrive, she is very open to persuasion attempts because fewer things are distracting her. Through their high mobility, wearable devices can be persuasive by offering **current**, **contingent** and **coordinated** information [14]:

- **Current Information:** If connected to a network, wearable devices can gather the most current information, which is also the most persuasive.
- **Contingent Information:** Wearables can take personal variables (e.g., position, heart rate, etc.) into account while trying to persuade.
- **Coordinated Information:** Wearable devices can share their information with other networked devices. This adaptability can make persuasion attempts more effective, as they can be executed on whichever device is currently used.

Challenges

While wearables are per se very well suited to provide persuasive experiences, they generally suffer from **bad long-term engagement**. Studies suggest that although many people buy a wearable device, they quickly abandon them. Research conducted by Endeavour Partners found that more than half of the Americans who own an activity tracker no longer use it [27]. An explanation for this is that users learn how to identify the measured data themselves: After a certain period of usage, an athlete with a fitness tracker will learn how much calories a typically 30-minute workout burns. After that, she might not need the fitness tracker anymore and stops wearing it.

Long-term engagement is significant for persuasive wearables, as behavior change can be a lengthy process. Ledger and McCaffrey identified three key factors to improve long-term commitment to wearables and attached services [27]:

1. **Habit Formation:** Wearables should encourage the formation of habits to sustain engagement. The formation can happen with the use of triggers on the wearable to remind the user to execute a specific behavior.
2. **Social Motivation:** Wearables can harness the power of social motivation to increase long-term engagement. Strategies include the sharing of achieved goals on social media platforms or the comparison with other users via high-score tables. Competing with others increases the likelihood of executing behavior changes.
3. **Goal Reinforcement:** Providing feedback mechanisms gives the user a feeling of progress towards her goal. Through their persistent presence, wearables can provide continuous feedback on goal achievement and the successfulness of the behavior change. This feedback will keep her engaged with the wearable and its persuasive power.

CHAPTER 3

Conception

3.1 Status and Approach

3.1.1 Status Analysis

In order to identify the shortcomings in the current state of the Stila application, it is necessary to perform a status analysis.

Current System

Stila comprises a heart rate monitor, an Android application, and a back-end server. Currently, only Fitbit fitness trackers are supported as heart rate monitors. Users connect their Fitbits via Bluetooth to their phones. Here the recorded heart rates can be synchronized with the Fitbit servers using the Android Fitbit app. After opening the Stila application, the user can log into her Fitbit account and download her heart rate data into the Stila application. Stila calculates the computed stress metric and displays a graphical representation. The user can also track her daily activities and feelings in the Stila Android application. Furthermore, the user can upload her heart rate and activity data to the Stila back-end server. In the Stila back-end, a supervised machine learning algorithm decides whether the experienced stress was more likely eustress or distress. The classifier learns from the labeled data provided by the tracked activities and emotions. Figure 3.1 shows a conceptional overview of the Stila system. The arrows symbolize the flow of data from one component to another. The main issue in earlier user studies was the absence of enough labeled data for the eustress/distress classifier: Users of the application are asked to record their daily activities and feelings regularly. This data is consequently used to train the classifier. It is advantageous to have a high amount of recorded activities because it improves the performance of the classifier. The results of an earlier study [17] show how infrequently the application was used to record activities: Six study participants used Stila for approximately one month. The study only resulted in 112 recorded activities. This means that on average less than one activity was recorded by each user per day. The reasons for the low engagement with the application were not investigated and can thus only be surmised:

Figure 3.1: Conceptional overview of the Stila system

Unintuitive Activity Tracking

Currently, users can only track their activities after they have completed them. If an activity like "working" takes several hours, the user has to remember the exact time when she started the activity and how she felt during it. This kind of activity tracking is rather unintuitive. Especially if the application is only opened once or twice per day, the user has to remember all her activities and feelings to record them. A more intuitive approach would be to let the user track her activities while she is doing them.

Cumbersome Activity Tracking

To track an activity the user has to remove her smartphone from her pocket or bag, unlock it, open the application and press the "Add new activity" button. This process might be too lengthy. Even saving a few seconds could dramatically improve the willingness of the user to track more activities.

Uninformed Users

Users of the Stila application do not get informed why they should record their activities. They are asked to invest time and effort in behavior they do not see the benefit for. The users should be made aware that the recording of activities improves the accuracy of the stress classifier and thus also benefits themselves.

Missing Feedback

Users of the application do not get enough feedback on their stress levels. The fact that heart rate data has to be synchronized to the Fitbit application hinders the feedback process. Usually, a user would only synchronize her data a few times a day. Therefore she does not experience real-time feedback, what results in less stress awareness.

Missing Reminders

Users can easily forget to track their activities because they are not reminded. It could have positive effects on the tracking frequency if this behavior was cued.

Poor User Experience

Many aspects of the user interface are suboptimal and result in poor user experience. For example, no custom activities can be created. When the user wants to track a custom activity type (e.g., "Reading") that is not included in the default selection, she has to enter the name in the interface. If she wants to track another incident of this activity type she has to enter the name again. No feature allows the saving of custom activity types.

3.1.2 Categorization and Approach

The problem of low user engagement shown by the small number of tracked activities shall be solved with the use of persuasive technology. The **target behavior** is defined as *Record Activities and Feelings in the Stila application*. It is important to improve the user experience in such a manner that the user will regularly open the application and track her activities.

Categorization in O'Brien's Process of Engagement

Analyzing the problem through the lens of O'Brien's Process of Engagement shows that a high number of **re-engagements** should be initiated. Prolonging the time between the point of engagement and disengagement does not achieve the desired goal: Typically users only record one activity per session, what likely cannot be changed by a deeper engagement. A frequent re-engagement of users, on the other hand, could drastically increase the number of tracked activities in the Stila application.

Categorization in the Behavior Grid

In Fogg's Behavior Grid the target behavior can be mapped as a **purple path** behavior: This means that the target behavior comprises the incrementation of an existing behavior. Furthermore, the behavior change should be permanent and not limited to a specific time span. Users of the Stila application typically track their activities less than once per day. This means that the target behavior already exists but is not performed often enough. Being a permanent behavior change, in this case, means that the tracking of activities should be performed from now on until the user deletes the application from her phone.

Analysis with Fogg's Behavior Model

Aforementioned problems show that the application is too hard to use. In Fogg's Behavior Model this is represented by the **ability dimension**. To cue the target behavior, it is necessary to increase the user's ability to perform it. It can be assumed that users of Stila are intrinsically motivated to use the application. They can get feedback on their stress levels and improve their stress awareness. This intrinsic motivation is very hard to manipulate, especially if the behavior change is to be permanent [13]. Thus, the approach chosen in this thesis concentrates on improvements in the ability dimension. Another shortcoming of the current application is the lack of **triggers**. After the ability has been improved enough to lift the target behavior above the action line, it has to be triggered. The trigger has to be executed in the right circumstances and at the right point of time.

Approach using Fogg's Functional Triad

As mentioned before, a way to manipulate the ability to perform a behavior change is to utilize Fogg's Functional Triad. The **tools** corner of the triad holds the means to make the target behavior easier, thus increasing the user's ability. It has to be examined how the technology involved in the Stila system can be changed to act as persuasive tools.

Wearables

In the context of this thesis, it seems appropriate to use **persuasive** wearables to achieve the target behavior. Smart watches are especially well suited for the task at hand because they frequently have built-in heart rate monitors. The Stila system shall be changed in a way that allows for the support of smart watches as a heart rate monitor and a wearable interface. Following the design guidelines in chapter 2.4 (*Wearables)* an application version for smart watches with a subset of the most important features shall be designed. The unique possibilities of wearables shall be utilized to further increase the ability of the user to perform the target behavior and trigger her at the right moment. Furthermore, the smart watch will act as a **cognitive** wearable, because it will determine the user's feelings (computed stress) based on biological measures (heart rate). An ordinary wearable will be transformed into a cognitive wearable by the addition of software.

In sum, the approach of this thesis is to improve user engagement by using a cognitive wearable as a persuasive tool. The increased user engagement will materialize as a higher number of tracked activities, which is expected to improve the stress awareness of the user.

3.2 Conceptional System Overview

Various conceptional changes are being made to the application to add persuasive capabilities. The most important aspect is the addition of a *Wear OS by Google* companion application, which can run on many modern smart watches. The wearable application is intended to be used in addition to the phone application. This section gives a short overview of the new Stila system by illustrating which tasks will be handled by which component. The features are described in more detail in chapter 3.3 *(Increasing Ability with Persuasive Tools)* depending on their persuasive tool type. Figure 3.2 shows a graphical representation of the new system concept. Depicted are only the smart watch application and the Android smartphone application components. The back-end will not be changed within the scope of this thesis. The smart watch takes over the task of heart rate measurement. This is achieved

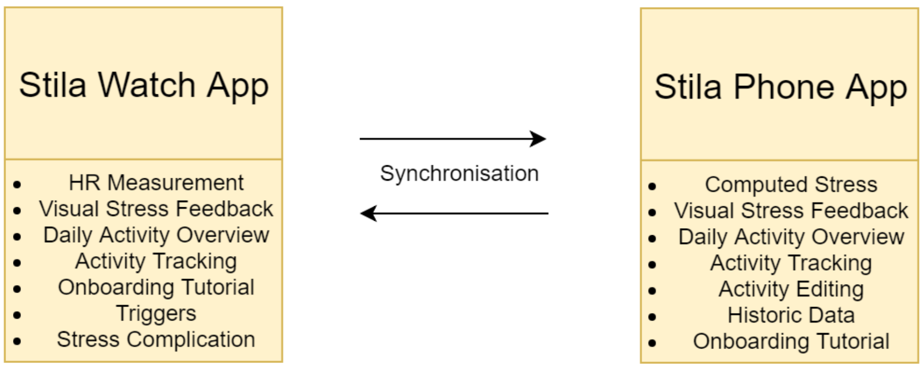

Figure 3.2: Conceptional overview of the Stila system after the addition of a watch application

by creating a Stila watch face that runs as foreground service and thus is able to measure the user's heart rate periodically. Graphical feedback on daily stress levels is available on both watch and phone. Likewise, the recording of the user's activity is possible on the watch and the phone. Both of the devices show an overview of the tracked activities on the current day. Historical data concerning the stress levels and the recorded activities is only available on the phone application. Due to the lacking of input mechanisms, the editing of recorded activities is also exclusive to the phone application. An onboarding tutorial that explains the features and intentions of the application is included in both of the apps. Furthermore, a trigger algorithm on the watch reminds the user at appropriate times to track her activities and feelings. The system allows the user to decide whether she wants to use a Wear OS smart watch or a Fitbit. If she chooses to use a Fitbit, the features exclusive to the smart watch are not available to her.

3.3 Increasing Ability with Persuasive Tools

This chapter describes how the ability of the user to perform the target behavior is improved with the utilization of persuasive tool types in a smart watch. The changed aspects of the Stila system are illustrated by showing the final implementation of the application. Details about the implementation are described in chapter 4 *(Implementation)*.

3.3.1 Reduction

Persuasive technology of the reduction type simplifies the target behavior [14]. In Stila the tracking of activities can be simplified in various ways:

Reducing Access Time

The access time of a device is defined as the time a user needs to initiate an interaction with it. This includes the time needed to acquire the device, e.g., from a pocket. Ashbrook et al. compared the access time for a personal digital assistant (PDA) to the access time of a wrist-worn device. He concluded that a user can interact with a wrist-worn device almost two seconds earlier [1]. These conclusions also translate to the Stila application: An interaction with the watch application is faster than an interaction with the phone application. This simplifies the handling of the application and favors the frequent tracking of activities. The actions needed to track an activity on the phone are as follows:

1. Remove phone from pocket
2. Unlock phone
3. Open the Stila application
4. Press the "Add new activity" button
5. Track the activity

The tracking of activities on the watch application, on the other hand, only consists of the following actions:

1. Flick wrist to look at the watch
2. Open Stila application
3. Press the "+" button
4. Track the activity

Figure 3.3 shows screenshots illustrating the way the user takes from the application launcher of the watch to the screen where she can track her activities: Not only is there one action less to do, but a flick of the wrist is also considerably faster than removing the phone from

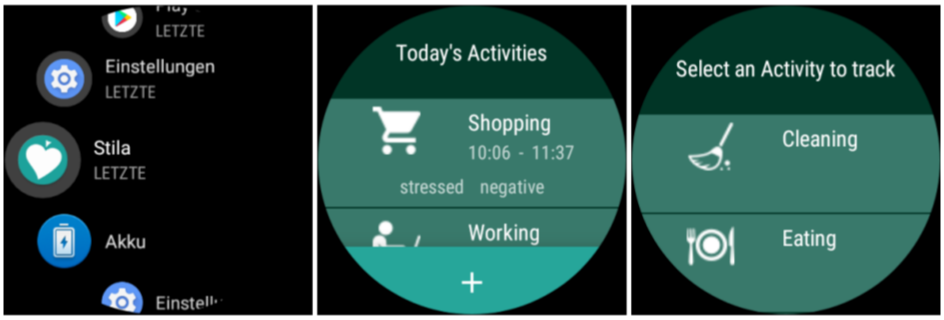

Figure 3.3: Screenshots showing the way the user takes when tracking an activity: Application launcher, main screen, and select activity screen

the pocket. Ashbrook et al. state that 78 percent of the access time is spent by retrieving the phone from its holder [1]. This time is greatly reduced by using smart watches.

Intuitive Time Tracking

As mentioned before, activity tracking in the Stila Android application is retrospective. A user tracks an activity after she has completed it. This means that she has to remember the time she started and the time she ended the activity, what makes it more difficult for her to track an activity. The Stila watch application allows simultaneous tracking of activities. When the user begins an activity that she wants to track, she can press the "Play" button. After she ended the activity, she presses the "Stop" button. Now she will be asked to specify her feelings and emotions during the activity. The application will automatically do the time tracking. If the user wishes to manipulate the tracked time (e.g., because she forgot to stop the activity), she can do so. This approach makes it easier for the user, as it requires less attention and also promises the gathering of more accurate temporal data. Figure 3.4 shows the watch application before (left), during (middle) and after (right) the tracking of the activity "working". If the user does not want to use concurrent time tracking but wants

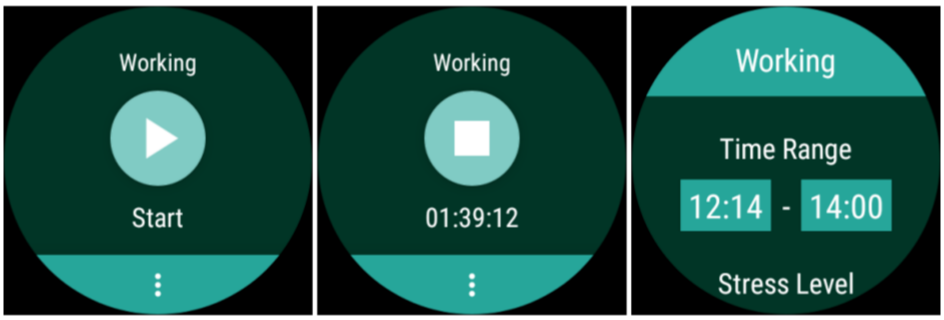

Figure 3.4: Screenshots of the Stila watch application showing intuitive time tracking

to record an earlier activity, she can also do this on the watch. A click on the menu icon (three points) takes her to a context menu where she can choose to track an earlier occurrence of the activity or cancel the interaction. Figure 3.5 shows this context menu. After she chooses the "Record earlier Working" option, she will be taken to the last screen of the time tracking dialog instantly. Here she can specify the start and end time of her activity.

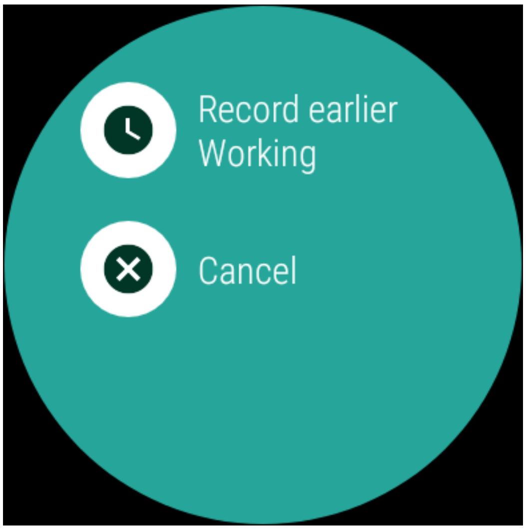

Figure 3.5: Screenshot of the context menu

Shortcut

A shortcut to the application's main screen has been created to further increase the ability of the user to track an activity. A long press on the watch face takes the user directly to the Stila application. This eliminates the necessity to open the watch's application drawer and search for the Stila application. With that the simplicity of the tracking feature has further been increased.

Device Support

By developing a Wear OS companion application for smart watches the application becomes available to a bigger audience. Previously only Fitbit devices with a heart rate sensor could be used. Now, smart watches by many manufacturers are supported. Both the *Fitbit* Android application and the *Wear OS by Google* Android application have each roughly 10,000,000 downloads in the Google Play Store [1]. This means that by supporting Wear OS devices the target audience doubles. The support of Wear OS devices not only opens the application up to a bigger target audience, but it also allows owners of both devices to use either of them. This decreases entry barriers for new users and increases the application's persuasiveness by simplicity.

3.3.2 Self-Monitoring

Persuasive capabilities of technology can be increased by adding self-monitoring features. These allow the user to check her progress in relation to the target behavior or an overall goal. The Stila watch application provides multiple self-monitoring tools:

Activity Overview

The activity overview (visible centrally in Figure 3.3) shows the user's tracked activities on that day. With that the application gives the user a quick overview of her progress relating to the target behavior. Furthermore, the overview increases the user's stress awareness

[1] https://play.google.com

by showing her tracked stress levels and whether the feeling was positive or negative. In the example depicted in Figure 3.3 it can be seen that the user was negatively stressed during a shopping trip. While a similar overview is also available via the phone application, it is much easier to access it in the watch application and is, therefore, more persuasive.

Stress Graph

The watch application includes a graphical representation of the user's measured computed stress levels. This can be seen in Figure 3.6. The self-monitoring capabilities of the stress graph do not support the target behavior directly. Nevertheless, it assists the overall stress awareness of the user and is therefore also counted as a persuasive self-monitoring tool. By using the stress graph, the user can quickly get an overview of her computed stress levels on this day. The graph is scrollable as well as zoomable.

Figure 3.6: Screenshot of the stress graph on the watch

Stress Complication

Another self-monitoring tool of the Stila watch application is the stress complication. Sullivan defines a complication as "[...] anything on the watch face that is not the time [...]" [56]. The Stila stress-complication is a small widget that shows the user's stress level directly on the watch face. It differs between three levels of stress: *Relaxed*, *neutral* and *stressed*. The levels are determined based on the newest computed stress value of the user. The stress-complication simplifies self-monitoring because the application does not have to be opened to gather stress feedback. The complication can be perceived in the periphery of the user's field of vision. Therefore, the user can gather information about her stress levels, while not being distracted by it. Figure 3.7 shows the Stila analog watch face with the stress complication showing the stress level "neutral".

Improved Accuracy

The watch application also enhances the self-monitoring capabilities of the watch application by improving the heart rate tracking accuracy. The previously used Fitbit devices do not allow a custom heart rate tracking interval. The measurements are triggered based on

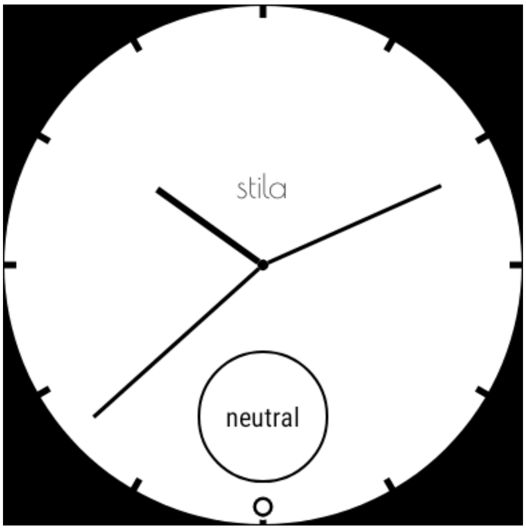

Figure 3.7: The Stila watch face with the stress complication presenting stress level "neutral"

Fitbit's internal heuristic every 5 to 10 seconds [21]. Especially 10-second intervals might not be enough for accurate heart rate variability data. The Stila watch application allows heart rate measurements in arbitrary multiples of one second. This means that the watch application can even be used to measure the heart rate every second, which allows for a far more accurate calculation of the heart rate variability. Better heart rate variability measurements produce more meaningful stress feedback for the user.

Real-Time Stress Tracking

A major advantage of the watch application over the usage of Fitbits is the ability to display real-time data. This is dependent on the system architectures: When Fitbit devices are used, the heart rate data has to be manually uploaded to the Fitbit cloud via the official Fitbit application. After that, the data must be manually downloaded to the Stila application, by logging into the user's Fitbit account and starting the synchronization. There is no way to do this automatically in the background. In the Wear OS application, however, the heart rate data can be streamed from the watch to the phone, where the computed stress calculation happens. Real-time tracking is an important aspect which greatly improves the stress awareness of the user. The user can instantly identify her stress level in a specific activity, rather than retrospectively making the connection between data and activity.

3.3.3 Tunneling

A tunnel is a predetermined sequence of screens a user passes through [14]. The tunnel offers an opportunity to persuade because in it the user normally gives the application her full attention. Three tunnels were added to the Stila application to create a more persuasive experience. A new user starts with the installation of the Stila application on the phone, so the tunnels on the phone are always experienced first.

Onboarding Tunnel on the Phone

After the user opens the Stila Android application for the first time, a text on the screen asks her whether she wants to use a Fitbit or Wear OS heart rate tracker. If she chooses to use a Wear OS heart rate tracker, she starts to go through the onboarding tunnel on her phone. This process involves ten steps. Each step of the tunnel explains a different aspect of the application and its purpose. It lowers the entrance barrier to the application by making concepts like *computed stress* easier to understand. It also illustrates why it is important to track activities. The user has to go through all of the steps to use the application. The onboarding concept simplifies the application by showing the user the benefits and intentions of the application before it is used for the first time. On exiting the tunnel, the user is asked to log in with her Google account. Before this, the onboarding dialog explains to the user why it is necessary to log in. This explanation can help to avoid a deterrent effect on privacy-concerned users. The tutorial messages are each enriched with an illustration. The illustrations make the presentation of the tutorial visually more pleasing and less tedious. Figure 3.8 shows all 10 screens of the tunnel. The texts displayed in tunnel are presented in the following:

> **Welcome to Stila**
>
> **Deal with your stress**
> Stila gives you feedback on your stress levels and helps you identify harmful stressors in your life.
>
> **Identify stress types**
> Positive stress helps you to perform at peak level when it matters. Negative stress harms your health and hinders your performance.
>
> **Companion application**
> To start using Stila, please also install the Stila Wear OS application on your watch.
>
> **Real-Time Tracking**
> Stila analyzes stress by measuring your heart rate on your Wear OS device.
>
> **Your Stress Diary**
> To increase the accuracy of the stress detection algorithm, you should periodically record your real-world activities and feelings.
>
> **Flexibility**
> You can track your activities either directly on your watch or in this application. They will be synchronized automatically across all your devices.
>
> **Calendar Integration**
> To make things easier for you, Stila also automatically downloads events from your Google Calendar.
>
> **Machine Learning**
> Upload your data in the Synchronization menu of the mobile application. Our machine learning algorithms will analyze your stress data in greater detail.
>
> **Done**
> Please now login with your Google Account. Enjoy Stila!

Figure 3.8: All screens in the onboarding tunnel of the phone

Usage Tutorial Tunnel on the Phone

The second tunnel a user enters in the Android application is the usage tutorial tunnel. In this tunnel, the most important features and their location are presented to the user. The application's screen darkens and highlights the location of each feature on the screen. Furthermore, a short tutorial text describes the highlighted features. The user can move forward in the tunnel by touching the screen. The tunnel highlights the stress graph, the location of the *Synchronize* button, and the location of the *Add new activity* button. After that, the user leaves the tunnel and can freely use the application. Figure 3.9 shows the last screen of the usage tutorial tunnel. The tutorial tunnel simplifies the interaction with the application by showing the user the most important features and their purpose.

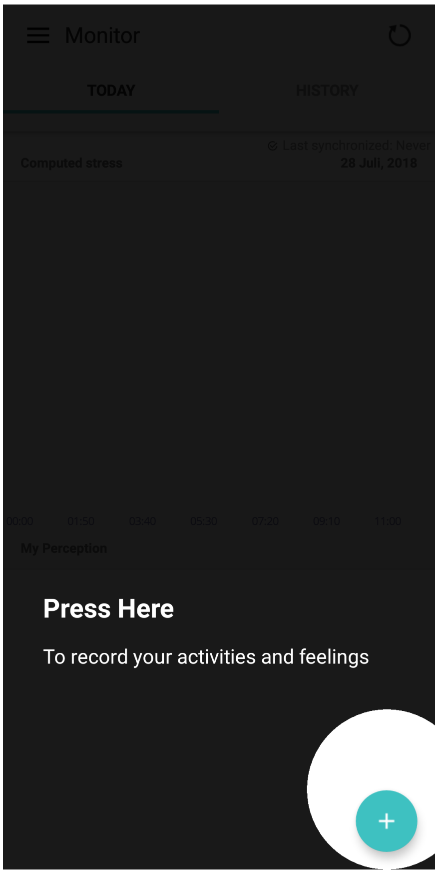

Figure 3.9: The last screen in the tutorial tunnel of the phone

Onboarding Tunnel on the Watch

During the onboarding tunnel on the phone, the user is asked to install the Stila Wear OS application. When the user opens the Wear OS application, she is again greeted with another onboarding tunnel. This tunnel's purpose is to explain the features exclusive to the watch application. These features include the stress complication and the shortcuts amongst others. At the end of the onboarding tunnel on the watch, the user is asked to change the watch face to a Stila watch face. This step is necessary to record heart rates for the stress calculation. After pressing a button, the user is taken to the Wear OS watch face picker, which is scrolled automatically to the Stila watch faces. This approach greatly simplifies the usage of the application, as the user can no longer forget to change the watch face or has to search for the watch face picker and the right watch face. Figure 3.10 shows three steps of the onboarding tunnel on the watch and the watch face picker.

Figure 3.10: Three screens of the watch's onboarding tunnel and the watch face picker

3.3.4 Tailoring

Tailoring describes persuasion through customization [14]. This refers to the personalization of information as well as to the shaping of software to a user's needs. Multiple features were included in the Stila application to allow customization:

Watch Face Types

To cater to all tastes, two types of watch faces were built: An analog and a digital watch face. Both watch faces have the same functionality and only differ in style and number of complication slots (see section *Customizable Complication Slots*). The user can switch between the watch faces in the Wear OS watch face picker. The different watch face types allow the user to customize the style of her wearable depending on her situation, outfit or personal taste. With that, the persuasive power of the smart watch application is increased and the hedonic qualities of the watch improved, which could lead to a better adoption rate. Figure 3.11 shows the Stila digital watch face on the left and the Stila analog watch face on the right.

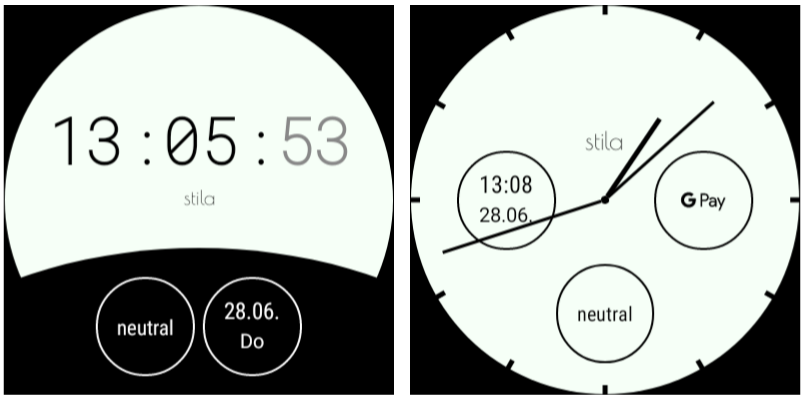

Figure 3.11: The digital and analog Stila watch faces

Watch Face Colors

To further increase the customization options regarding the watch faces, the color of the background and other elements are freely selectable. A color-picker has been added to the *Settings* menu of the watch application which allows the user to define hue, saturation, and brightness of the background color. Figure 3.12 shows the color-picker on the left. The user can also change the color of the watch arms or digits (depending on the watch face type). Here the user can only switch between white and black elements. This limitation helps to keep the watch face readable when using different colors: E.g., when a very light color is chosen for the background, black arms or digits have to be used to ensure enough contrast. Figure 3.12 shows a dark blue watch face with white arms on the right.

Figure 3.12: The background color picker and the Stila analog watch face with a dark blue background

Customizable Complication Slots

As mentioned in previous sections, complications can be used to increase the informational content of a watch face. The Stila watch faces posses customizable complication slots, which let the user display many different kinds of information. Because of design reasons, the analog watch face has three complication slots while the digital watch face has only two. The user can change the complications in the settings menu of the watch application. Supported complications include the Stila stress complication, a calendar complication, a weather complication amongst many others. Figure 3.13 shows the complication picker for both the analog and digital watch face. The user can add or change a complication by clicking on one of the plus icons on the screen. The analog watch face with three active complications (date, weather, and burned calories) can be seen in figure 3.12. Complications provide a huge increase in the customizability of the watch face. Users can display the information that is most relevant to them. This ensures that the user is content with the Stila watch faces and uses it daily to track her stress levels.

Accuracy Customization

Another tailoring aspect of the new Stila system is the addition of customization options in the measurement and synchronization of data. The watch application allows the user to select how often the heart rate sensor on the back of the watch is activated. This customization lets the user decide whether she wants more accurate data (short measurement

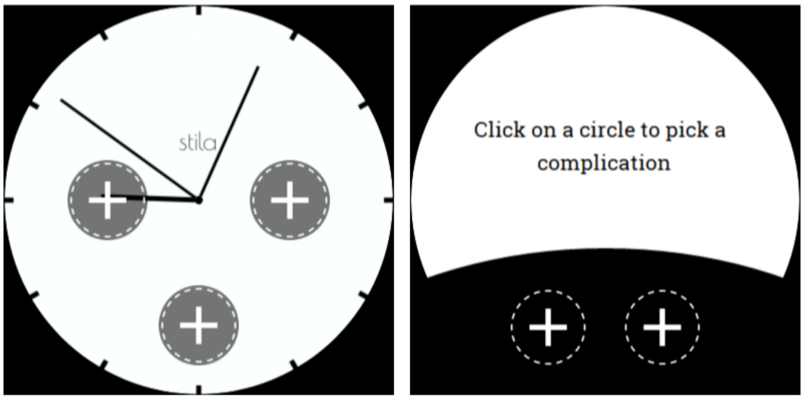

Figure 3.13: Complication pickers for the Stila watch faces

intervals) or save battery power (long measurement intervals). The user can also deactivate the heart rate measurement on this screen. Figure 3.14 shows the heart rate measurement settings screen on the watch. The user can also change the interval in which the watch communicates with the phone. A longer communication interval saves battery power, while a shorter interval provides more accurate computed stress data on the watch. These accuracy customization features let the user tailor her user experience to her needs and current situation.

Figure 3.14: The heart rate measurement settings screen

Custom Activity Types

Another way the user can personalize her experience is by creating custom activity types. In this context *activity type* describes what activity the user is doing: Reading, working, cleaning etc. In previous versions of the application, it was not possible to easily track custom activity types. If the user often performs an activity that is not included in the predefined activities like "practicing the piano", she would have to enter the activity name every time she wants to record it. By introducing custom activity types, this is easier to do: The user only has to create a new custom activity once and can then continue to use it, every time she wants to record the activity. Now, the user does not have to type the name of the activity when tracking it. To fit into the design scheme of the predefined activities, the user can also select an icon for her new custom activity. Custom activity creation is limited to the phone application because the watch does not have the appropriate input capabilities. Figure 3.15 shows the screen a user sees when adding custom activity types on the phone application. Custom activity types simplify the interaction with the application by removing the need to type activity names. Depending on the personal usage, this can be a very significant improvement in the user's ability.

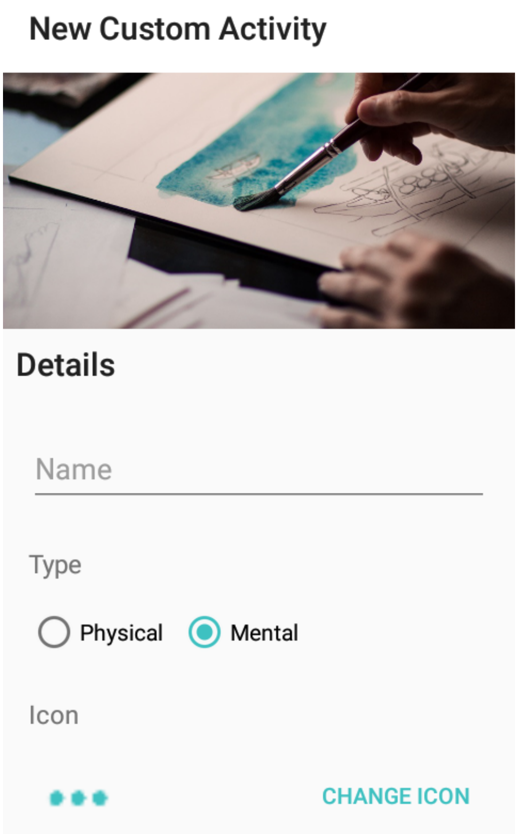

Figure 3.15: The screen that allows the creation of custom activities

3.4 Triggers

Besides motivation and ability, triggers are the third part of Fogg's Behavior Model [13]. The Stila Android application at the start of the thesis did not have any triggers. Because of the approach of this thesis that includes the mapping of Fogg's Behavior Model onto the Stila application, triggers had to be implemented. Without the presence of triggers, it can not be examined whether the changes increased the ability of the user. As mentioned before, Fogg's Behavior Model requires the user to have motivation, ability and to be triggered at the same time to perform the target behavior. Without triggers, the target behavior likely will not be performed. This shows, that while the goal of this thesis is to increase the user's ability, the addition of triggers is also necessary. Triggers were added to the smart watch application as well as to the smartphone application (in Fitbit mode) to achieve a higher verifiability of this thesis's goal.

Triggers in Mobile Technology

In the context of Stila, the function of a trigger is to remind the user to track her activities regularly. The apparent technical implementation of a trigger in applications for smart devices is a **push notification**. The push notifications that were added to the Stila system are **facilitator triggers**. Facilitator triggers additionally simplify the target behavior [13]. In Stila, this is implemented by providing a shortcut from the notification to the screen that lets the user record her activities.

Timing

An important aspect when working with triggers is to find the right point in time. A trigger should be executed when the audience is both motivated and able to perform the target behavior. It is challenging to measure these attributes with the help of technology. However, in the case of Stila as a smart watch application, it is possible to estimate the ability of a user based on her current stress level. When a user is stressed, she is probably doing something that requires her full attention. Therefore it is highly unlikely that she has the ability to track her activities. If the user is not stressed the chance that she has a high ability is far greater. To compensate for inaccuracies in the assessment of the current ability of the user, the notifications have a "snooze" function. This button can be pressed, when the user cannot track her activity at this moment but wants to be reminded again in 30 minutes.

Frequency

Currently, there is no way to measure the motivation of a person or build heuristics for this. Consequently, there will be some triggers that will come at the wrong time. Therefore it is important to design the triggers in such a way that they are not unnerving to the user. This can be implemented by only sending push notifications at certain times of the day and by limiting the frequency of notifications. If a trigger is sent too often or on wrong occasions, they can have a negative effect on the target behavior. This should be prevented where possible.

Because the system architecture of Stila in Fitbit mode is different to Stila in Wear OS mode, the realization of triggers slightly differs. In the Wear OS application, the unique advantages of persuasive wearables are utilized.

3.4.1 Triggers in Fitbit Mode

The application sends a notification on the smartphone if these conditions are true:

- The user has already opened the application on the same day at least once
- No activities were tracked in the last two hours
- The time is between 9am and 9pm
- The application was not opened in the last three hours

All these conditions together ensure that a receptive user is triggered, but an unmotivated or unable user is not annoyed. Because the application cannot collect real-time data in Fitbit mode (see above), it is not possible to identify moments of low stress and therefore high ability. The chosen values for the number of hours are arbitrary. Future work could include the manipulation of the trigger frequency.

Figure 3.16 shows a notification on the phone in Fitbit mode. After the user clicks on the notification, she is taken to the main screen of the application, where she can add an activity.

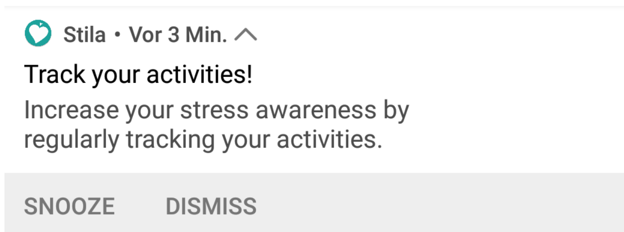

Figure 3.16: A Stila notification in Fitbit mode

3.4.2 Triggers in Wear OS Mode

Triggers in Wear OS mode are multi-faceted. The watch application was enriched by adding **smart notifications** and **backup notifications**. Both of these notification types are only visible on the watch.

Smart Notifications

Triggers that take the stress level of the user into account were named "smart notifications". These notifications take advantage of the real-time capabilities of the smart watch application and the increased persuasiveness of wearables. A smart notification informs a user that she was stressed in a particular time frame and asks her if she wants to track her stressful activity. This should increase the engagement with the trigger because the user has a clear benefit from recording the activity that stressed her. Whether the algorithm decides to send a smart notification is dependent on several factors. All of the following conditions have to be true for a smart notification to be sent:

- The user is wearing the watch and is recording her heart rates
- The user's stress level is *neutral* or *relaxed*
- Before the current stressless phase there were at least 20 stressful minutes

- No activities were tracked in the last hours or are being tracked now
- The last notification is older than one hour
- The time is between 9am and 9pm

These conditions are again arbitrarily chosen to ensure that the user is not annoyed by too many notifications.

Backup Notifications

So-called "backup notifications" were added to the watch application to make sure, the user is regularly triggered even if the conditions of the smart notifications are never met. If a user is not stressed for the whole day, she would never be triggered to record her activities. The backup notifications are thus sent as a fail-safe to ensure the user is triggered. The backup notifications are sent under the following conditions:

- The user is wearing the watch and is recording her heart rates
- No notification was sent in the last three hours
- No activities were tracked in the last two hours or are being tracked now
- The time is between 9am and 9pm

Figure 3.17 shows an exemplary smart notification on the left next to a backup notification on the right.

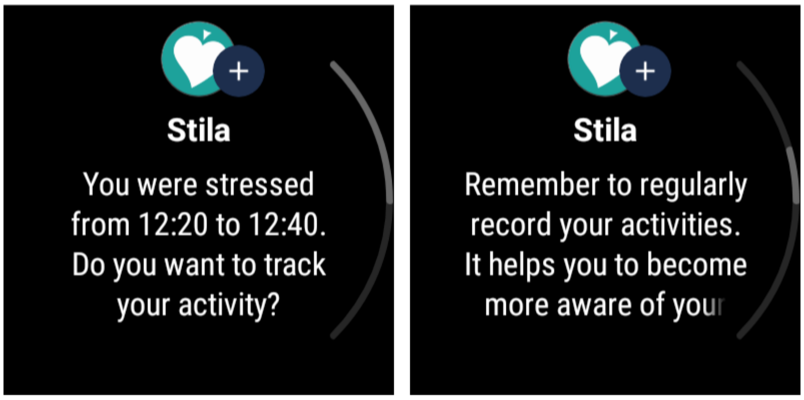

Figure 3.17: A smart notification (left) and a backup notification (right)

The full text of the smart notification is as follows:

> "You were stressed from 12:20 to 12:40. Do you want to track your activity?"

The text of the backup notification is the following:

> "Remember to regularly record your activities. It helps you to become more aware of your stress."

After the user clicks on one of the notifications, she can scroll through the whole text and interact with the notification by either clicking on the *Record Activity* action or by opening the context menu. The context menu holds two further interaction alternatives: The *Snooze* and the *Dismiss* action. The full notification with a hidden context menu can be seen in

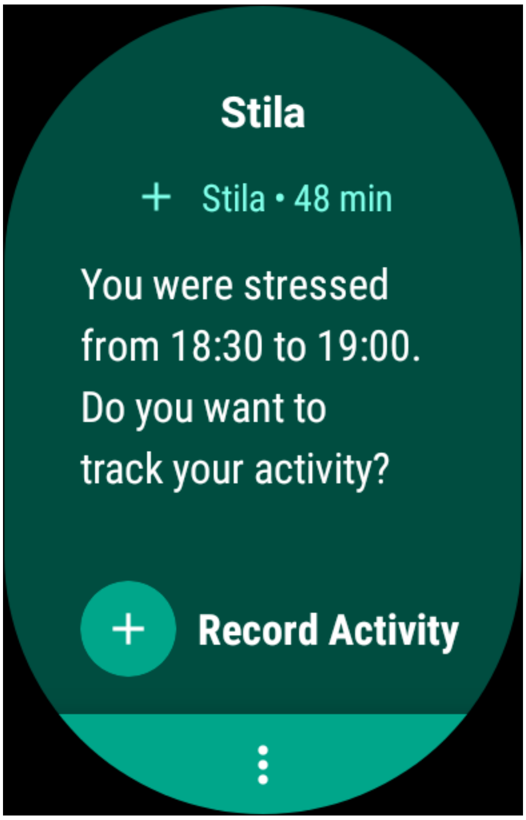

Figure 3.18: Scrollshot of the full smart notification

Figure 3.18. If a user clicks on the *Record Activity* action, she will be redirected to the *Select Activity* screen of the watch application. Here she can pick an activity she wishes to track. On the following screen, she can record her feelings and the precise period of the activity. If a smart notification initiated the interaction, the recognized start and end of the stress phase is entered in the interface as the time-frame of the activity. This feature further simplifies the recording of stressful phases. Note, that this feature could also have been classified as a **Suggestion Tool** in Fogg's Functional Triad.

3.5 Interface Design

This section describes all interface design decisions that were not discussed in the previously. These decisions were made in consideration of the design advice collected in chapter 2.4.4 (*Design and Aesthetics*).

3.5.1 Wear OS Design Guidelines

As mentioned earlier, it is advantageous to rely on design guidelines if available. Google provides design guidelines for the development of Wear OS software [8]. The most important factors are summarized in the following:

- **Provide timely information:** The right information should be given at the appropriate time.

- **Design glanceable interfaces:** Interfaces should be kept uncluttered and easy to read. Information should be structured in a clear hierarchy.
- **Make elements easy to tap:** Clickable targets should be well-spaced and big enough to tap. Text input should be avoided were possible.
- **Create time-saving applications:** Interaction flows in Wear OS applications should help users to complete tasks quickly.
- **Use vertical layouts:** Switching between horizontal and vertical layouts should be avoided as it confuses the user.
- **Integrate data through the lens of time:** Because the interface runs on a watch it is advisable to display data with a time dimension if possible. This can be done by displaying how data changes over time rather than presenting absolute values.

These guidelines are taken into account in addition to the design advice identified in the theoretical framework.

3.5.2 Navigation

Navigation is an important aspect of every application. In the context of this work, navigation describes the means a user has to switch screens in the application. Figure 3.19 and Figure 3.20 show the navigation flow of the Stila watch application. Figure 3.19 depicts navigation possibilities originating from the *Activities* screen, while Figure 3.20 shows other elements accessible from the *Graph* and the *Settings* screen. User actions are symbolized by rectangles with rounded corners and screens of the application are depicted by rectangles with sharp corners. Users can navigate between the screens in the direction of the arrows by performing the respective action. In most cases, arrows can be traversed in the other direction by swiping from left to right on the display.

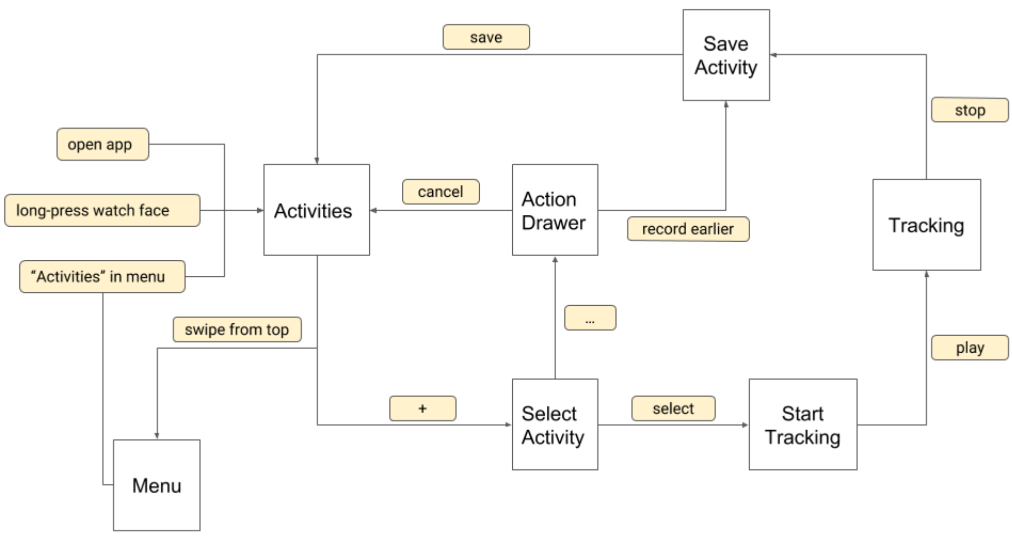

Figure 3.19: Navigation flow originating from the activities screen

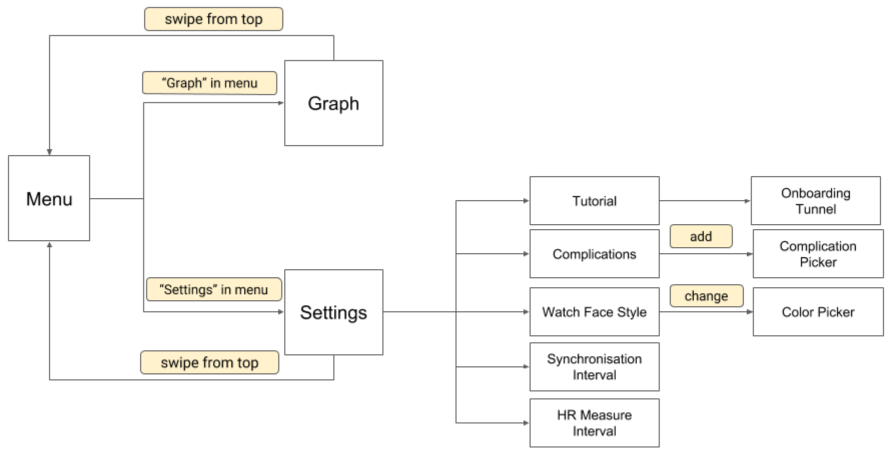

Figure 3.20: Navigation flow originating from the graph and settings screen

The Stila watch application uses three navigational tools: A **Navigation Drawer**, an **Action Drawer** and **Swiping Gestures**.

Navigation Drawer

The navigation drawer is a menu that slides down from the top of the interface. It contains hyperlinks to the three main navigational sections of the application: The *Activities* screen, the *Graph* screen and the *Settings* screen. In each of these screens, the navigation drawer can be accessed by swiping from the top to the bottom of the display. Figure 3.21 shows the navigation drawer after it has been swiped down by the user. The three icons each link to one of the aforementioned main screens of the application. The screen that is currently visited by the user is highlighted using a light circle. The highlighting helps the user to keep her orientation. In figure 3.21 the user is currently residing on the *Activities* screen. After the user clicks on one of the icons, she is taken to the corresponding screen, and the navigation drawer minimizes.

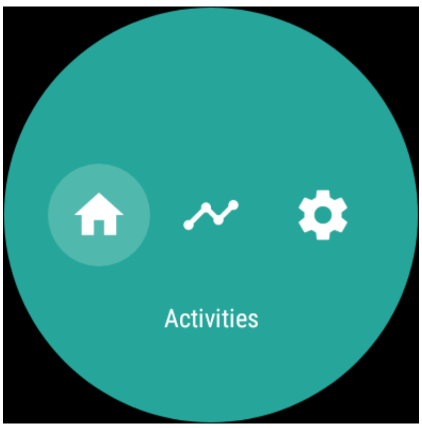

Figure 3.21: Navigation drawer

Action Drawer

The action drawer is similar to the navigation drawer. It holds user-actions contextual to the current state of the application. Unlike the navigation drawer, it can be accessed by swiping from the bottom of the screen. The action drawer is not available on all screens but only on the *Activities* screen, the *Current Activity* screen and while reacting to notifications. The other screens do not require contextual actions and thus have no action drawer. Different icons symbolize the action drawer. On the *Activities* screen, it bears a plus icon and takes the user directly to the *Select Activity* screen. On the *Current Activity* screen and in notifications the action drawer is symbolized by three dots, which open the drawer when clicked. The contextual actions of the *Current Activities* screen can be seen in Figure 3.22.

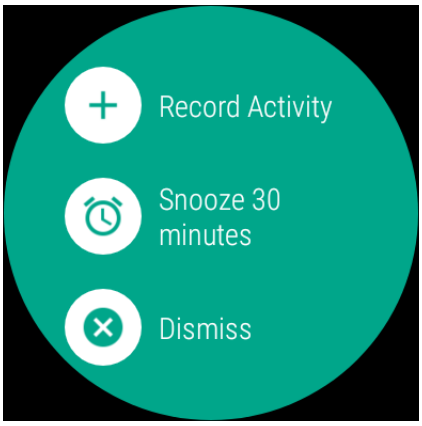

Figure 3.22: Contextual actions of the action drawer in a Stila notification

Peeking

To inform the user of the availability of the navigation and the action drawer, they "peek" into the interface when the *Activities* screen is opened. "Peeking" means that the screen is overlayed with a small part of the drawers. When the user scrolls the activity overview, the preview of the drawers disappears to allow the user to see the whole screen. Figure 3.23 shows the peeking navigation and action drawer.

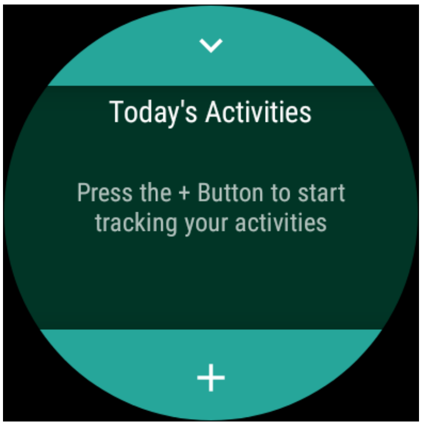

Figure 3.23: Peeking drawers after the application is started

Consistent Positioning

The navigation drawer and the action drawer can always be found in the same position if available. This conforms with Shneiderman's [53] guideline to keep the position of elements consistent over all screens.

Quick Returns

The Stila watch application also achieves conformity with the design advice to allow quick returns to the previous screen and the watch face. A finger swipe from left to right on any screen takes the user to the previous screen in the history stack. Furthermore, the user can return to the watch face by pressing the hardware button, which is available on all Wear OS devices.

3.5.3 Interaction

This section describes how user input is facilitated on the Stila watch application and how it differs from input on the Stila phone application.

Interruption Handling

As stated in the theoretical framework of this thesis, smart watch applications have to be designed in such a way that allows users to be interrupted while using them. If a user is distracted during the tracking of an activity and closes the application, the current screen is saved. The state is also saved when the operating system switches to standby because the interaction timed out. A timeout happens when the user does not interact with the screen for approximately 15 seconds. Wear OS also triggers a timeout when the watch is moved in a manner that suggests that the user is no longer looking at the screen. If the user opens the Stila application again, the activity tracking screen on the history stack is opened, and the start time is still saved. This enables the user to utilize her watch in other ways while tracking activities with the intuitive time tracking feature.

Text Input

To make the Stila Wear OS application easier and less frustrating to use the interface was designed without the need to input text on any screen. The naming of custom activity types can be done solely on the phone application. The custom activity types are then synchronized to the watch application. This design decision follows the Wear OS guidelines which state that text input should be avoided were possible [8] and the design advice to limit functions from the theoretical framework of this thesis.

Activity Tracking

A centerpiece of the application is the recording of the user's daily activities. Several changes were made, to adapt this interaction from the larger screen of a phone to the small screen of a watch. Due to the smaller display, the screen on the application has to be scrolled much more. Additionally, all elements are oriented vertically on the watch because there is not enough width to orient them horizontally. Furthermore, the following changes were made:

- The field that lets the user track the time was moved to the top of the screen. It is the most important feature and should, therefore, have the most prominent place.

- To keep the interface more consistent, all activity dimensions that are tracked by sliders are placed before the activity dimensions that are tracked by radio buttons. This helps the user to fill out the form more quickly because she does not have to switch between input modalities as often.

- The *none* option in the activity dimension *drug intake* was moved to the middle of the radio button group because it resembles the neutral position of the variable.

- The activity icon is not displayed on the screen of the watch because it would take up unnecessary space.

- The ability to indicate whether an ability is physical or mental was removed in the Wear OS application. This dimension still can be changed when creating a new activity type on the phone. This dimension does not ever change for an activity type and can be cut from the watch application without consequence.

- The ability to change the day of the tracked activity was not implemented on the watch. Users of the Wear OS application can only track activities that happened on the current day. Older activities have to be tracked via the phone application. Again, this is a feature that is less used and contradicts the design guideline to provide timely information and only make the essential features available.

- Some space was saved by omitting the *neutral* label in the sliders. These are not essential for the understanding of the input mechanisms.

- A field that lets the user add notes to the activity was not implemented in the watch application. This feature was assessed as not essential and would be difficult to use, due to the limited input modalities of the watch.

- The *CONFIRM* button has been replaced by a large image button depicting a save icon on the watch. The *CANCEL* button has been omitted. This helps the user to quickly scroll through the dimensions she does not want to track and conclude the interaction. To cancel, the user can swipe from left to right on the display of the watch.

Figure 3.24 shows a screenshot of the activity tracking screen on the phone and a scrollshot of the same screen on the watch.

Stress Graph

Due to the small screen of the watch, it is challenging to present a graphical representation of the variation of the user's stress levels. To overcome this problem Shneiderman's **Visual Information-Seeking Mantra** has been implemented. As mentioned before, Shneiderman [53] advises that information visualizations should present an overview of the data first, then allow to zoom or filter and provide details only on demand. Due to the rather simple nature of the presented stress data, neither a filter function nor details-on-demand were implemented. When the user opens the graph, it is zoomed out and shows roughly the last three hours. The user can zoom in the graph by using pinch gestures and scroll the graph by dragging it.

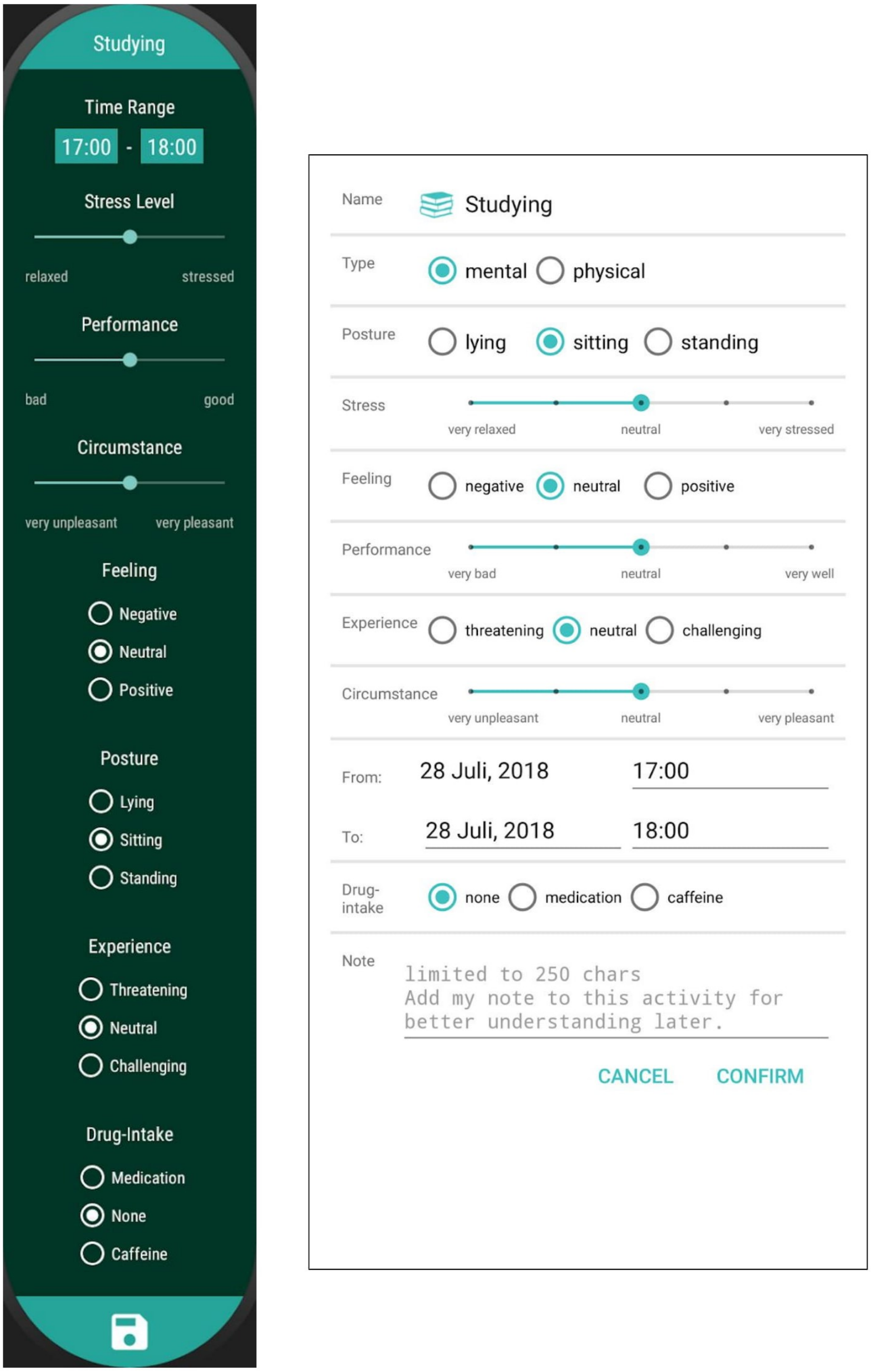

Figure 3.24: Activity tracking on the watch (left) and on the phone (right).

3.5.4 Aesthetics

This section describes the aesthetics of the watch application's interface design.

Color

The design advice by Shneiderman [53] to use color conservatively and consistently has been followed in the development of the watch application. Furthermore, the Wear OS design guidelines [8] state, that a darker color palette should be used to save battery power. Figure 3.25 shows the different color palettes of the phone and the watch application. Both applications use a teal color palette. However, for the watch much darker colors were used. The watch application uses white text, while the phone application uses black text, to achieve a high contrast to their respective palettes.

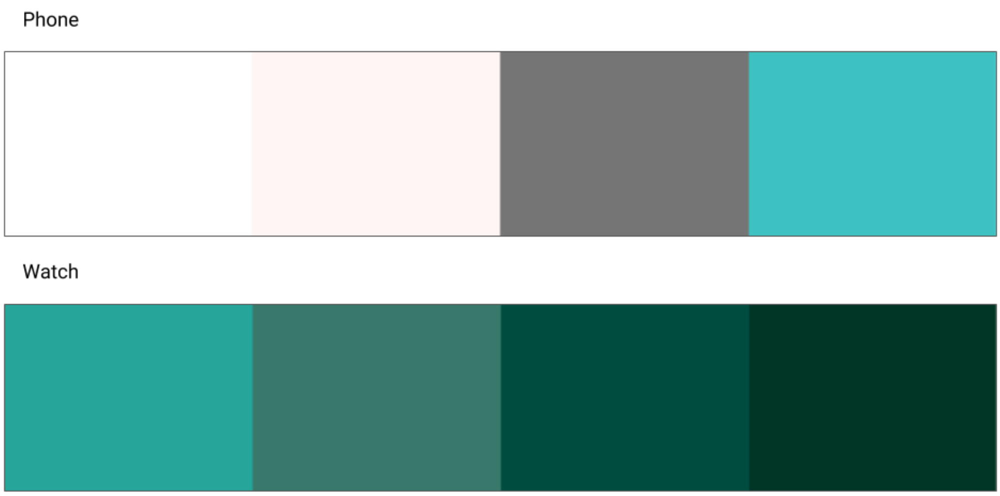

Figure 3.25: Color palettes of the phone application (top) and the watch application (bottom)

Typography

The Wear OS guidelines were also followed in terms of typography. The guideline advises to use the *Roboto Condensed* font, which is an ancestor of Google's standard font *Roboto* [8]. Text written in *Roboto Condensed* takes up slightly less space, which is favorable when designing for small displays.

Watch Faces

The design of the Stila watch faces was largely determined by aesthetic reasons. Watch faces are by their nature very passive interfaces because they do not allow much interaction. Therefore a designer of watch faces has a high level of creative freedom. In the case of Stila, the digital and analog watch faces have been designed to appear as classical as possible. Bold design choices were avoided. The watch faces should appeal to as many users as possible. Both watch faces use a very clear design with limited design elements, laying the focus on the display of time. This makes the watch face highly glanceable and appropriate for everyday use. In order to save battery life, watch faces need to support Wear OS's *ambient* mode (also called *always-on* mode). The watch switches from *interactive* to *ambient* mode when it is not interacted with for a certain amount of time. The watch continues to show the time but goes into an energy saving state. The Wear OS design guidelines [8] require watch faces to fulfill several properties when running in ambient mode:

- Use a limited color palette
- Use a black background
- Avoid large filled areas
- Only update once a minute

These requirements ensure not only the saving of battery but also avoid burn-in effects on certain displays. The Stila watch faces meet all of the aforementioned guidelines: On entering the ambient mode, the background color is switched to black, and the digits or arms are displayed in white. The seconds arm on the analog watch face and the seconds counter on the digital watch face are omitted, because Wear OS updates watch faces in ambient mode only once every minute. The design of the digital watch face requires a change in the layout of the digits to ensure the interface is still symmetric after removing the seconds counter. Figure 3.26 shows the analog watch face in interactive mode on the left and in ambient mode on the right. The digital watch face in interactive and ambient mode can be seen in Figure 3.27.

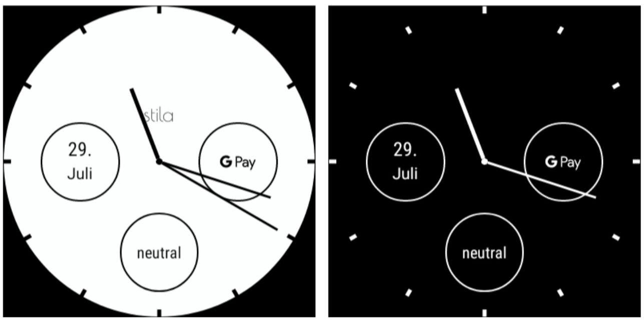

Figure 3.26: Analog watch face in interactive (left) and ambient mode (right)

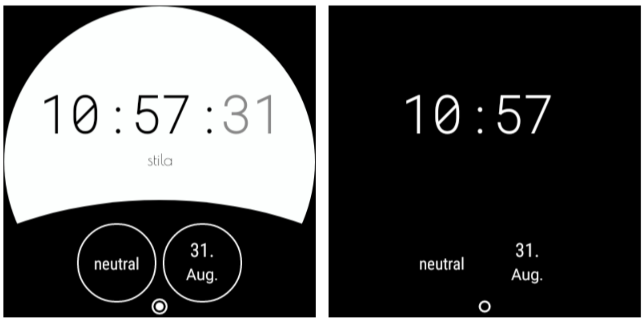

Figure 3.27: Digital watch face in interactive (left) and ambient mode (right)

CHAPTER 4

Implementation

This chapter describes the technical implementation of the aforementioned persuasive features. For brevity's sake, the following sections concentrate on the most important and interesting concepts.

4.1 Wear OS

Wear OS or as it was previously known **Android Wear** is an operating system for smart watches and other wearables. It is based on Android for mobile phones and can be described as its wearable flavor [37]. Wear OS is developed and maintained by Google. The first version was released in June 2014 and featured a subset of functions of Android 4.4. In the first year of its existence 700,000 devices running Wear OS were sold [30]. A Wear OS smart watch usually "augments the smartphone and provides a simpler and lighter user interface that allows the user to receive notifications and address trivial online interactions, in a less intrusive manner" [37]. Wear OS is thus not aimed at independent smart watches but at devices which are tethered to a smartphone by Bluetooth or Wifi.

All Wear OS devices also have got a microphone which enables them to respond to the "Ok Google" hot word that starts a voice-based interaction with the *Google Assistant* [37]. Wear OS was chosen for this project because it provides a common abstraction layer and an application programming interface for smart watches of various manufacturers. These include *Huawei*, *Motorola*, and *Asus*. An advantage of a commonly used OS is that applications only have to be written once and can be run on many devices.

Software

The programming of applications for Wear OS is very similar to the programming of applications for Android. Most libraries and system features can be used on Wear OS as they would be on a traditional Android Device. Exceptions to that include the *android.webkit* API which is used to display web pages and the *android.print* API which offers methods to print files [37]. These libraries would not be practical on wearable devices and are thus not part of Wear OS. Apart from that, it is necessary to adapt the design of the application to work with a smart watch form factor [58].

Software for Wear OS can take two forms: Watch faces and Apps: **Watch faces** are services that run in the foreground of the watch if no application is currently active. The main purpose of watch faces is to show the time. However, a watch face is not limited to that. A developer has almost complete freedom in deciding which information a watch face should display. Developers create watch faces by programmatically drawing on a Java Canvas. Canvases can display arbitrary graphics such as lines or polygons. The usage of canvases makes the creation of watch faces highly flexible but also very complex, as all elements of a watch face like hands or the dial has to be drawn to the canvas programmatically.

Wear OS **Apps** are independent applications running on the watch. Wear OS Apps can function in two different ways: As a stand-alone application or as a companion application, which can only be used if another version of the application is installed on a connected smart phone. Stand-alone Apps were introduced in Android Wear 2.0 (later renamed to Wear OS) and also work in connection with Apple's iPhones [48].

Test Device

A *Huawei Watch 2* was used for the development and the initial testing of this application. This smart watch has a round 1.2-inch display and a relatively strong 410mAh battery, which promises up to two days battery life with "normal" usage. The device runs on a Qualcomm Snapdragon 2100 processor with four cores and a clock speed of up to 1.2 GHz. The watch has 768 MB RAM available. These specifications are roughly comparable to the Motorola G, a Smartphone from 2014. This shows that the Huawei Watch 2 is quite powerful considering its small size. At the time of testing, the watch ran Wear OS 2.10 (based on Android 8), which was the newest available version.

Figure 4.1: Front view of a Huawei Watch 2 with an analog watch face [24]

Development

The development of Wear OS apps is similar to traditional Android applications. The code is mostly written in Java (though Kotlin is also possible). Google recommends the usage of *Android Studio* which is an IDE specialized on the development for Android and Wear OS. While in development, applications have to be transferred from the PC to the watch. The application can either be pushed directly from a PC to the watch via a USB cable or by connecting an Android phone to the PC and using the phone's Bluetooth antenna. It is also possible to use emulators to test Wear OS applications. This is generally not recommended because applications for smart watches should be tested as they are used: On a very small screen worn on the wrist of the user. The emulator furthermore does not support the collecting of sensor data such as heart rates.

4.2 System Overview

This section gives an overview of the Stila system in a more concrete way. It shows the underlying database model, as well as the code structure of the Stila applications.

4.2.1 Structure

As mentioned before, the Stila system comprises three conceptional components: A heart rate monitor, the Stila smartphone application, and the back-end. The user can choose to use the system with a Fitbit or with a Wear OS smart watch as a heart rate tracker. Depending on whether the system runs in Fitbit or Wear OS mode the architecture differs: In Wear OS mode the Stila Android application is extended by a wearable **companion** application running on the smart watch. In Fitbit mode, no wearable application is available. The following sections primarily show the application in Wear OS mode as most of the described features are not available in Fitbit mode.

Technical Structure of the System

The technical setup of the Stila system in Wear OS mode is outlined in the following: The watch measures the heart rate of the user with the help of one of the Stila watch faces. The Stila watch application sends the collected heart rates with their timestamps to the Stila phone application. The communication is realized by using the *Wearable Data Layer API* (see chapter 4.5 *Communication and Data Synchronization*). After the heart rates and the timestamps are synchronized with the phone, they are used to calculate the computed stress metric. Following the successful calculation, the results are sent back to the watch. This communication enables the system to show a graphical representation of the stress levels on both watch and phone. Activities can be created on the watch or on the phone. Both devices show all created activities, no matter if the activity was tracked on the phone or the watch. The phone application contains a sign-in interface that can be used to log into a Google account. The Google account simplifies the creation of user profiles and is used to synchronize the user's data with the back-end.

Code Structure

The relevant code of the Stila system is structured into three modules. A module is a container for code, which can be reused in other projects or other modules by adding it as a dependency. The three modules of the Stila system are named **app**, **wear** and **common**:

- **app:** The app module contains all code that is only used by the phone application.
- **wear:** The wear module holds code that is exclusive to the watch. This includes the watch faces, heart rate measurement algorithms and communication code.
- **common:** The common module contains code, that is used by the wear and the application module. Examples include system-wide constants and handlers for databases that are available on both phone and watch. The common module is added to the app and wear module as a dependency.

4.2.2 Databases and Shared Preferences

The Stila Android application and the Stila Wear OS application use **SQLite** databases to manage and persist most of its data. SQLite is a relational database engine which is commonly used in mobile applications for Android and iOS [12]. The watch and phone application each have their own set of databases. Most tables exist on both of the devices. Nevertheless, they are not synchronized automatically and thus can have different contents. Figure 4.2 shows all relevant database tables and their attributes on both devices. Tables depicted in red are exclusive to the phone, while blue tables are only present on the watch. The tables are not connected to each other by the use of foreign keys. The managed data did not make foreign keys necessary. Furthermore, a database without relations between tables can be easier to maintain and is in some cases less error prone. The following section briefly describes the purpose of each relevant table in the databases [1]:

- **Activity:** This table saves all activities tracked by the user. This includes all recorded dimensions and a timestamp for the start and the end of the activity. The activity table exists on the watch and phone and has to be synchronized.
- **ActivityType:** The ActivityType table saves all activity types such as “Reading”, “Working” and so on. Each activity type can be categorized as “physical” or “mental”. Furthermore, the id of an icon can be persisted, which is used to enrich the frontend of the applications.
- **DeletedActivity:** This table is used to facilitate reliable communication. It contains all activities that were deleted by the user.
- **Heartrate:** The measured heart rates and their timestamps are saved in this table.
- **HRV:** The HRV table contains calculated heart rate variability (HRV) scores and computed stress values together with their timestamp. It is, e.g., used to graphically present the stress intensity of the user.
- **OidcToken:** This table holds the authorization information for the Google sign-in.
- **Profile:** The Profile table contains information about the user, including her email and Google Id. It is used to identify uploaded information in the back-end.

[1]Some names were slightly altered in this thesis to make them conform with SQL naming conventions and more readable

- **RunningActivity:** This table is exclusive to the watch. It saves the name and the start time of the activity that is currently tracked by the user.
- **Token**: Fitbit authorization tokens are saved in this table.

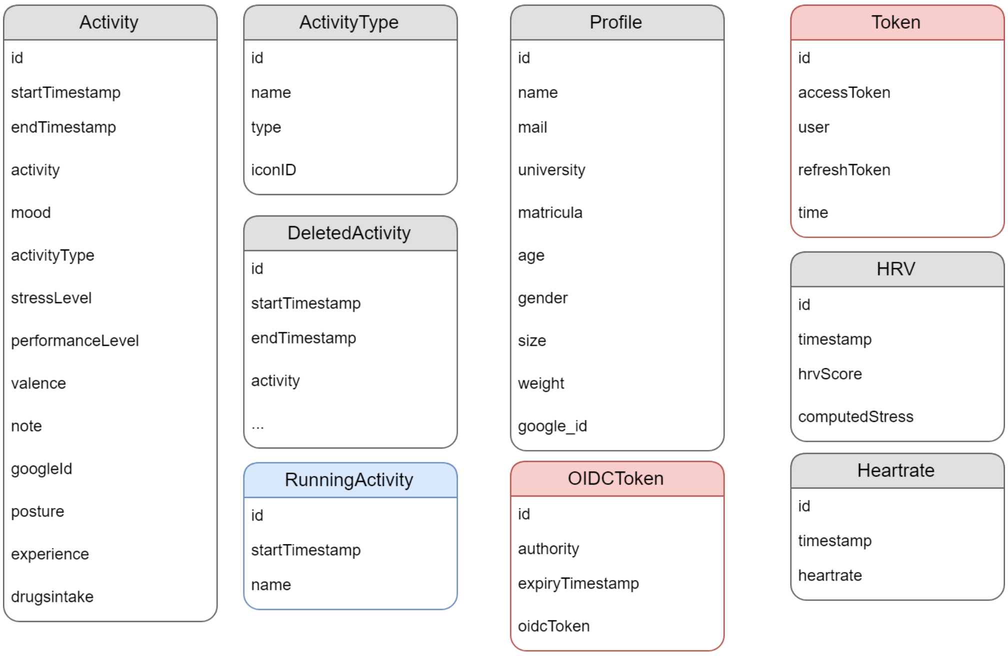

Figure 4.2: Simplified diagram of all relevant database tables of Stila

Furthermore, both applications each use a *SharedPreference* file to persist arbitrary internal data as key-value pairs. The SharedPreference file in the watch application includes information about the watch face type, the background color, measurement and synchronization intervals amongst others. The phone application persists whether the application runs in Fitbit or Wear OS mode and if the application is started for the first time in its SharedPreference file. The SharedPreference files are not synchronized between the devices.

4.3 Heart Rate Measurement

In order for the system to periodically analyze the user's heart rate variability (see chapter 4.6 *Stress Computation*), it is first necessary to gather raw heart rate data. Most current Wear OS smart watches have an optical heart rate monitor. These monitors shine light onto the skin and measure its reflection to determine the change of blood volume under the skin. From this information, cardiac cycles can be recognized and counted. This technique is called *photoplethysmography* (PPG) [51].

The measurement has to be started regularly to perform accurate stress calculations. This makes it necessary to run an Android service on the Wear OS device that can periodically initiate new heart rate measurements. Background services in Android are not reliable enough to implement periodical measurements: The operating system often stops these services or prevents them from starting to save energy. To counteract this problem, Stila uses watch faces to measure the heart rate. Watch faces are foreground services which run as long they are used. Being a foreground service allows the application to read the heart rate sensor whenever necessary and in a reliable manner. Because the Stila watch face always runs in the foreground of the application, the OS will not stop it for to energy optimizations.

Figure 4.3 holds a UML activity diagram that shows the heart rate measurement process. The three main components are one of the **Stila watch faces**, the **HeartRateScheduler**, and the **HeartRateListener**. All of these components are represented by their own Java classes. In this example, the **StilaDigitalWatchfaceService** is used, but the analog watch face acts

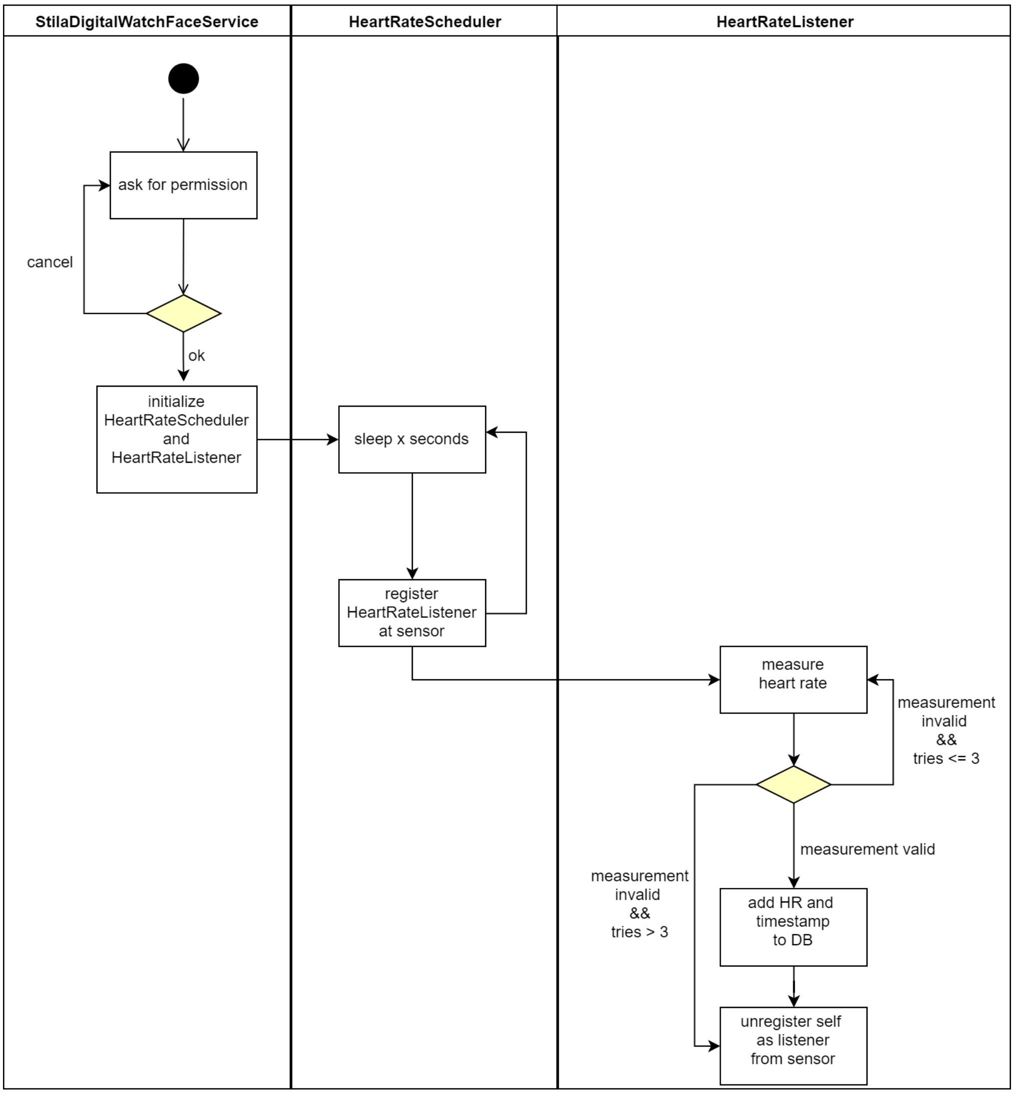

Figure 4.3: UML activity diagram showing the logic behind the measurements of heart rates by the watch

in the same way: After the Stila digital watch face is started, the service asks for permission to use the heart rate sensor of the watch. If the user accepts the permission request, she can start using the watch face. If she denies the permission, an error message is presented to her.

The watch face starts the **HeartRateScheduler** Runnable, in a new Java thread. The HeartRateScheduler's *run* method lets the thread sleep for however long the interval of two heart rate measurements should be. The interval can be changed by the user in the settings. E.g., if the user sets the interval for 10 seconds, the HeartRateScheduler sleeps for 10 seconds before waking up. After the thread wakes up, the HeartRateScheduler adds the HeartRateListener to the watch's heart rate sensor. A reference to the heart rate sensor can be acquired by using the system's SensorManager. After the HeartRateScheduler added the HeartRateListener to the device's sensor, it goes to sleep for the user-defined number of seconds.

The **HeartRateListener** inherits the *onSensorChanged* method from its superclass **SensorEventListener**. This method is called after the device's hardware sensor returns a heart rate measurement. The HeartRateListener now checks if the measurement was accurate enough by reading the returned SensorEvent's accuracy attribute. If the accuracy is not high enough (e.g., because the watch is not worn correctly) the measurement gets discarded, and the sensor tries to perform a measurement again. After three unsuccessful measurements, the HeartRateListener gives up and unregisters itself from the sensor. If the sensor measured successfully, the heart rate (in beats per minute) and the time (as an epoch timestamp) are persisted to the Heartrate database. After that, the HeartRateListener removes itself as a listener from the device's sensor. The HeartRateListener will be registered at the sensor again after the specified interval elapsed.

The user can also specify a continuous measurement of her heart rate in Stila's settings menu. In this case, the HearRateListener never gets removed from the sensor and measures the heart rate as often as possible. This is usually once per second. The HeartRateScheduler is also not running in this case because it is not needed.

A sleeping (background) thread is not the most power efficient solution for the HeartRateScheduler. An Android **AlarmManager** which wakes the CPU after a specified alarm interval was also tested for the heart rate measurement. While this solution seemed more power efficient, it did not work on all Wear OS devices. Some watches blocked the AlarmManager from executing the measurement code because it woke the CPU too often. A background thread, on the other hand, worked on all tested devices even though it is less power efficient.

4.4 Watch Faces

4.4.1 Class Organization

As mentioned in the previous section, the purpose of the Stila watch faces is primarily to facilitate periodic heart rate measurements reliably. The *Google Playstore* allows the packaging of Wear OS apps with watch faces. Therefore, the user does not have to install the Stila watch faces separately. The analog and the digital watch face are each controlled by their own services: The **StilaAnalogWatchFaceService** and the **StilaDigitalWatchFaceService**. To be able to act as a watch face in Wear OS, both of the services extend the abstract system class **CanvasWatchFaceService**. Because both watch faces generally perform the same tasks, they are abstracted by the use of an interface. The **StilaWatchFace** interface defines all methods a watch face has to implement in order to be used by the Stila system. When a watch face service is referenced in other parts of the application, it is allocated as a StilaWatchFace and not as a StilaDigitalWatchFaceService or StilaAnalogWatchFaceService. This abstraction layer helps to reduce code duplication because the same logic can be used, no matter if the analog watch face or digital watch face is active. The interface can also be used to extend the Stila system with more watch faces in the future. Figure 4.4 shows a simplified UML class diagram of the classes involved in the creation of the Stila watch faces. A stronger inheritance between the watch faces is not possible: Due to the different design, the logics differ greatly.

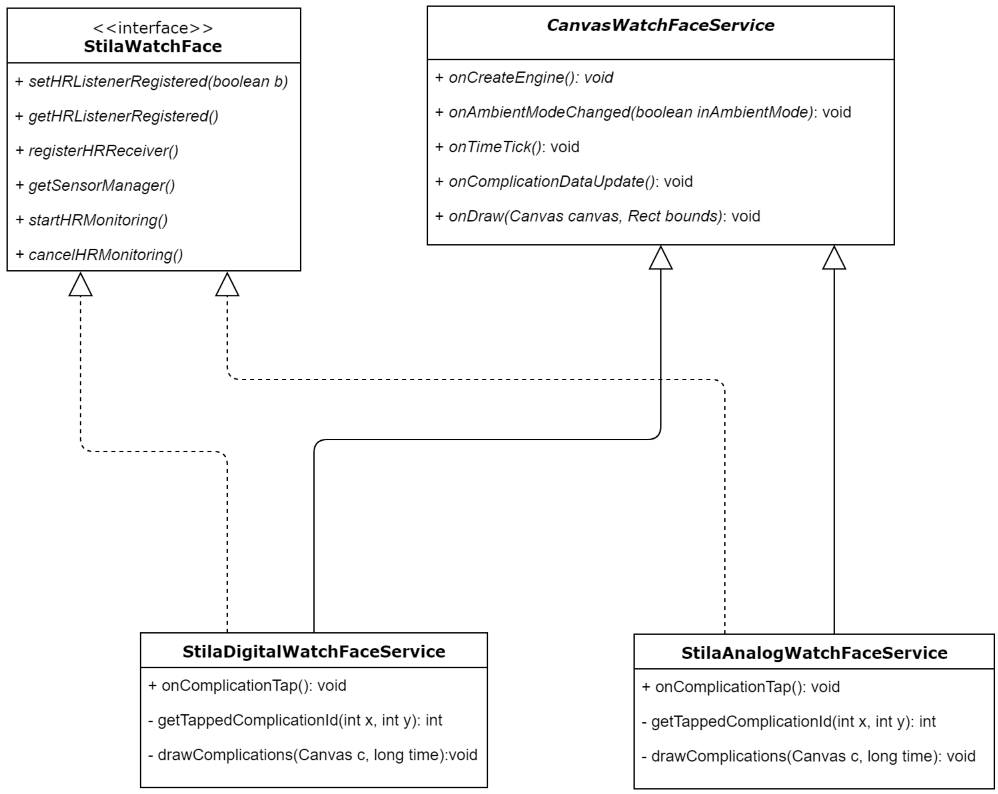

Figure 4.4: UML class diagram showing the class organization of the Stila watch faces

Registration in Manifest

To be recognized as a watch face, the relevant class has to registered in the application's manifest in a special way. Listing 4.1 shows the registration of the digital watch face in the watch's application manifest.

```
1 <service
2     android:name=".watchface.StilaDigitalWatchFaceService"
3     android:label="@string/digital_name"
4     android:permission="android.permission.BIND_WALLPAPER">
5     <meta-data
6         android:name="android.service.wallpaper"
7         android:resource="@xml/watch_face" />
8     <meta-data
9         android:name="com.google.android.wearable.watchface.preview"
10        android:resource="@drawable/preview_digital" />
11    <meta-data
12        android:name="com.google.android.wearable.watchface.preview_circular"
13        android:resource="@drawable/preview_digital_circular" />
14    <meta-data
15        android:name="com.google.android.wearable.watchface.
    wearableConfigurationAction"
16        android:value="lmu.pms.stila.CONFIG_DIGITAL" />
17    <intent-filter>
18        <action android:name="android.service.wallpaper.WallpaperService" />
19        <category android:name="com.google.android.wearable.watchface.category.
    WATCH_FACE" />
20    </intent-filter>
21 </service>
```

Listing 4.1: Registration of the digital watch face in the application's manifest file

The permission to bind a wallpaper in line 4 is needed because a watch face in Wear OS is the counterpart to a wallpaper on an Android phone. In line 6 the StilaDigitalWatchFaceService is registered as a wallpaper in the system. To inform the operating system that this service is a watch face and not an ordinary wallpaper an intent-filter is set in line 17. From line 8 to line 13, the preview images of the watch face are linked. This enables the watch to show a preview image of the watch face in the Wear OS watch face picker without starting the service. To have fitting images for both circular and rectangular watches, it is necessary to specify a rectangular and a circular preview image separately. The metadata statement from line 14 to line 16 implements the one-click shortcut from the watch face to the Stila application. This statement describes which action should be performed when the watch face is pressed. Usually, this is used to link a configuration activity to a watch face. In Stila, the main activity of the wear application is opened. The Stila main activity defines an intent-filter, which listens to the *CONFIG_DIGITAL* action. When the watch face is pressed, the *CONFIG_DIGITAL* action is performed and the main activity starts.

4.4.2 Watch Face Logic

After the watch face service is started, the *onCreateEngine* method is called. Here, resources like colors and fonts are initialized. The colors for the background and the digits or arms of the watch are read from the SharedPreference file. The user can manipulate this file in the settings menu. The settings are made independently from the chosen watch face and are used by both watch faces. This means that when, e.g., a user picks a white background color, both the analog and the digital watch face will be displayed with a white background.

The centerpieces of the watch faces are their *onDraw* methods that paint the watch face. This method is called by the *onTimeTick* method which invalidates the canvas periodically. In interactive mode, the canvas is invalidated every second to show the current second. In ambient mode, the canvas is only redrawn once every minute. The digital watch face in interactive mode has an additional timer mechanism, which invalidates the canvas every 0.5 seconds, while simultaneously hiding and displaying the seconds counter. This generates a flashing effect, which is familiar from traditional digital watches. This timer is not active in ambient mode or the analog watch face.

The *onDraw* method reads the system time in epoch milliseconds with the *System.currentMillis* method and converts it to a human readable time string in the user's timezone. It then draws the Stila design on the watch face canvas. This step differs largely depending on which Stila watch face is used. In the digital version, the time is written as a String on the watch face, in the analog version the arms have to be rotated in order to show the correct time.

The complications are managed by the *onComplicationDataUpdate* method which is called by the system when a complication has new data available. The painting of the complication to the canvas is performed by the *drawComplications* method which gets called by the *onDraw* method. This method renders the complications and its container on the watch face.

The switching between ambient and interactive mode is managed by the *onAmbientModeChanged* method. This method gets called by the system if the watch goes into standby. The colors of the watch face are then changed to the ambient color scheme (white elements on black background), and the canvas is redrawn.

4.5 Communication and Data Synchronization

The following section concentrates on the communication and synchronization of data in the Stila system.

4.5.1 Overview

Synchronized Data Types

It is necessary to communicate and synchronize certain data to use the Stila watch application as a companion to the phone application. As mentioned before, several database tables exist on both devices and thus have to be synchronized. The following tables have to be communicated between the devices:

1. **From phone to watch:**
   - **HRV:** The HRV and computed stress calculation (see chapter 4.6 *Stress Computation*) is performed on the phone and has to be sent back to the watch to allow the display of stress graphs, the stress complication, and smart notifications.
   - **DeletedActivity:** User-tracked activities can be deleted on the phone. To keep the activity overviews of the devices synchronized, the deleted activities have to be sent to the watch.
   - **Profile:** The profile contains the Google ID of the user. This ID is set while logging into the app on the phone. It is also needed on the watch to facilitate the collection of usage analytics on the watch.
   - **ActivityType:** Custom activity types can only be created on the phone. In order to use the activity types on the watch as well, it is necessary to send them to the watch.
2. **From watch to phone:**
   - **Heartrate:** The measurement of heart rates is performed by the watch. The heart rates have to be sent to the phone, to calculate the computed stress level.

3. **Two-sided:**

    - **Activity:** Activities can be tracked on the phone and the watch. To display the same activity overview on both devices, they have to be synchronized.

Wearable Data Layer API

To facilitate communication between wearable devices and phones the **Wearable Data Layer API** is used, which is part of the Google Play Services. The API is pre-installed on Wear OS devices and can be added to an Android phone by installing the *Wear OS by Google* application from the Google Play Store [37]. The Wearable Data Layer API offers a network that connects all Android devices (wearables and phones) of a user. Every device is represented by a **node** in the Wearable network. The communication between the nodes can happen via Bluetooth or the internet: A smart watch can send and receive data directly to or from a connected smartphone via Bluetooth tethering. It is also possible to communicate via Wifi and the internet. If no Bluetooth connection can be established, but both watch and phone have a connection to the internet, they can still communicate via the Wearable Data Layer. In this case, the data is sent to a Google Play Services cloud node and then synchronized to the other device. This functionality is hidden to the programmer: The Wearable Data Layer API automatically decides which route is taken to sent the data.

The Wearable Data Layer API supports two means of communication, which are each represented by its own API: The **DataAPI** and the **MessageAPI**. These mechanisms have different use cases:

- **DataAPI:** The DataAPI synchronizes data between nodes of the Wearable network. Arbitrary data can be *put* into the network and will be sent automatically to all nodes via the Wearable Data Layer. Other nodes can then *get* the data, manipulate it and *put* it in the network again. Other listening nodes will be notified about the changes in the synchronized data element. A synchronized data element is represented by the **DataItem** class in the DataAPI, which offers wrappers for several other data types. The DataAPI is very well suited for reliable synchronization of data between nodes: If a node is offline while another node manipulates the synchronized data, it will be informed about the accumulated changes once it goes online again. The DataAPI also supports two-sided manipulation of a synchronized data item between nodes. A synchronized DataItem can be seen as a network-shared object which can be written and read by all nodes of the wearable network [37].

- **MessageAPI:** The MessageAPI allows transient messages between nodes. A message with an arbitrary payload can be sent from one node to another. If the target node listens for this kind of messages, it can receive the message and read its content. Messages do not have guaranteed delivery. If the target node is offline while the source node sends the message, it is lost and will not be repeated automatically [37].

In sum, the DataAPI is more suited when a guaranteed synchronization of data items is desired. The MessageAPI should be used when the information that is sent does not require a delivery guarantee and can be repeated in the future.

Receiving of Data in the Stila System

The centerpiece of the communication between watch and phone in the Stila system are the **DataLayerListenerServices**: These classes exist in the watch module and the phone module. They extend the **WearableListenerService** which is part of the Wearable Data Layer

API. Communication that is sent from the phone is received by the DataLayerListenerService of the watch and vice versa. The DataLayerListenerServices override two methods of the WearableListenerService: The *onDataChanged (DataEventBuffer dataEvents)* method and the *onMessageReceived (MessageEvent messageEvent)* method. This service enables them to listen for DataAPI and MessageAPI events in the wearable network. The implementation of these methods differs from phone to watch.

The DataLayerListenerServices identify the content of an event by looking at its **path**. Paths are arbitrary Strings that are set while sending the data item or message. Because the paths have to be identical on watch and phone, they are stored in the **common** module in the class **CommunicationConstants**.

After a change of a shared data item was detected in the *onDataChanged* method, the synchronized data item (an array of JSON objects) is unpacked and further processed depending on the event's path. A received message is identified in the *onMessageReceived* method by reading its path. After that, the message and its payload is processed further.

Sending of Data in the Stila System

The code responsible for the sending of data is not bundeled in one class like the code for the receiving. Instead, the synchronization of certain data types is being facilitated by **Synchronizers**. In the Stila system, there are three synchronizers which all use the DataAPI to communicate. The **ActivitySynchronizer**, the **HRVSynchronizer** and the **ProfileSynchronizer** which are part of the commons module and can be used in either communication direction. The Synchronizers contain methods to send data structures as JSON arrays to other devices and are called by the application logic when necessary.

Heart rates are not communicated with a Synchronizer: Because the heart rates can be synchronized in a user-defined interval, they are sent using a **JobService**. A JobService is a mechanism that executes a defined method at fixed intervals. The class **HeartrateSyncJobService** is started by the operating system in intervals specified by the user (e.g., every 10 minutes) and sends the heart rates to the phone.

The MessageAPI only plays a subordinate role in the Stila system. It is used for auxiliary tasks and helps the implementation of more complex synchronization algorithms. The logic to send messages is bundled in the **MessageToPhoneUtil** class on the watch and in the **MessageToWatchUtil** on the phone.

Figure 4.5 shows a UML class diagram with all classes relevant to the communication in the Stila system with their most important methods and the module they are in. The inheritance relationship between Stila's classes and library classes are omitted for better readability.

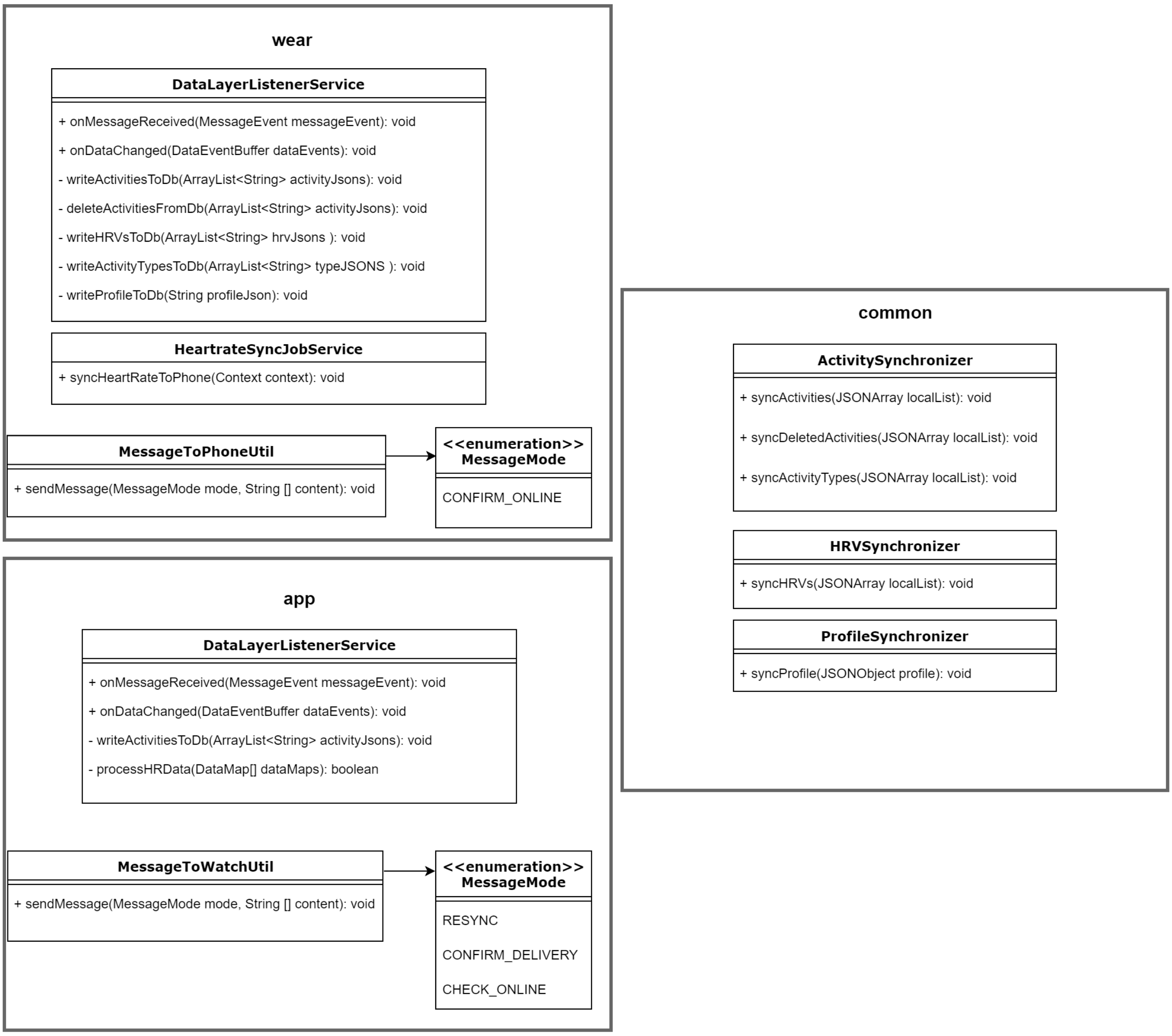

Figure 4.5: UML class diagram showing the classes responsible for the communication in their modules

4.5.2 Synchronization Mechanisms

The following section details the data synchronization using two examples: The communication of activities and the sending of heart rates between the devices. All other data types are synchronized similarly.

Activities

As mentioned before, activities can be added on the watch and the phone. This creates the need for a solid synchronization mechanism. The algorithm has to be able to synchronize the devices in various situations. E.g., it is possible that the user creates an activity on the watch and another activity on the phone while the devices are offline. After the connection is reestablished, watch and phone have to resynchronize to have the same set of activities in their database.

Figure 4.6 shows how the activity synchronization is performed in the Stila system. The depicted situation occurs when a user adds an activity on the phone. This activity now has to be synchronized to the watch. First of all, the application reads the last 100 saved activities from the DB and saves it as a JSON array. This number is arbitrary and signifies the number of activities that can be tracked on one device before it has to be synchronized with its companion. If the user tracks more than 100 activities before synchronizing, older activities can be out of sync. The ActivitySynchronizer's *syncActivities* method is called with the last 100 activities in JSON format. The Synchronizer creates a new **PutDataMapRequest** with the JSON array as payload and enters the *ACTIVITY_PATH* (a system-wide constant) as the path. If the phone is online or connected via Bluetooth to the watch, the JSON array containing the activities is broadcast into the Wearable Network. If not, the PutDataMapRequest will be send the next time the phone goes online. Now the JSON array containing the activities is shared across the Wearable Network.

The DataLayerListenerService on the watch consequently registers a change in the data under the *ACTIVITY_PATH*. It extracts the JSON Array containing the activities from the DataItem provided by the DataEvent. Now, the watch adds all activities to its database that were not already added. This means activities created on the phone will be added to the watch's database. If no new activities were added, the synchronization is finished. If new activities were added, the watch starts to synchronize its last 100 activities with the phone app. The watch completes the same steps as the phone did before. This second synchronization ensures that the databases are synchronized even if activities were added on both devices while they were not connected. The phone now receives all activities added by the watch in its DataLayerListenerService and runs through the same logic as the watch before. This goes on until both databases are in sync, which is identified by the fact that no new activities were added to the DB after the shared array has changed.

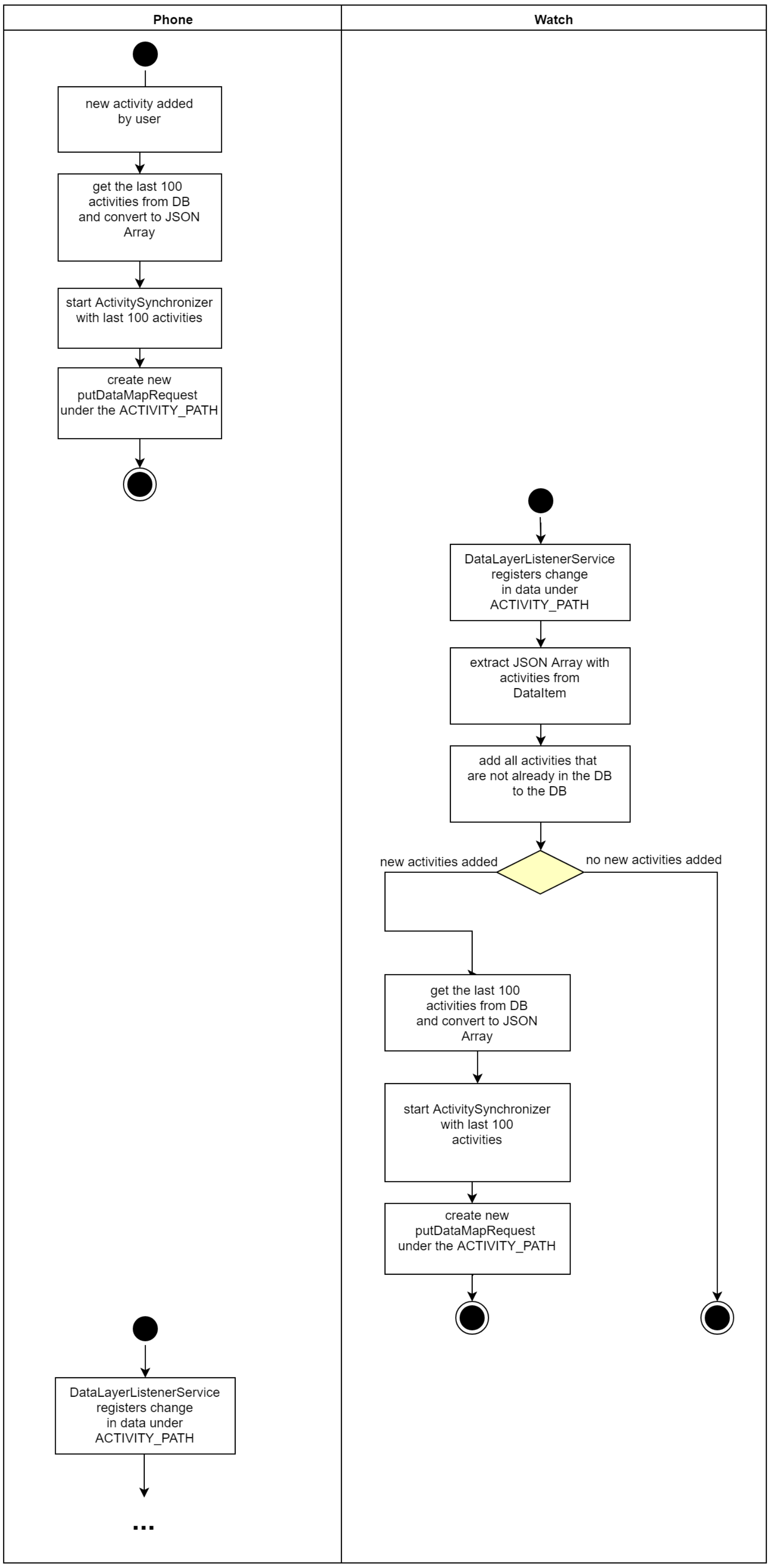

Figure 4.6: UML activity diagram showing the synchronization of activities from the phone to the watch

Figure 4.7 shows the synchronization of the databases of the watch and the phone in a conceptual way. Activities are identified by numbers, which show in what order the activities were tracked. Rectangles represent the devices' databases and the shared array in the wearable network. Each line represents a time step. The activities that are new for a component in a certain time step are colored green.

In the depicted situation, the watch and the phone were disconnected. Meanwhile, the user added activities on the phone and the watch. After the devices reestablish a connection, the activities tracked on the watch have to be sent to the phone and vice versa. In this example, the synchronization is started by the phone. The graphic shows how the activities are synchronized with the algorithm explained above.

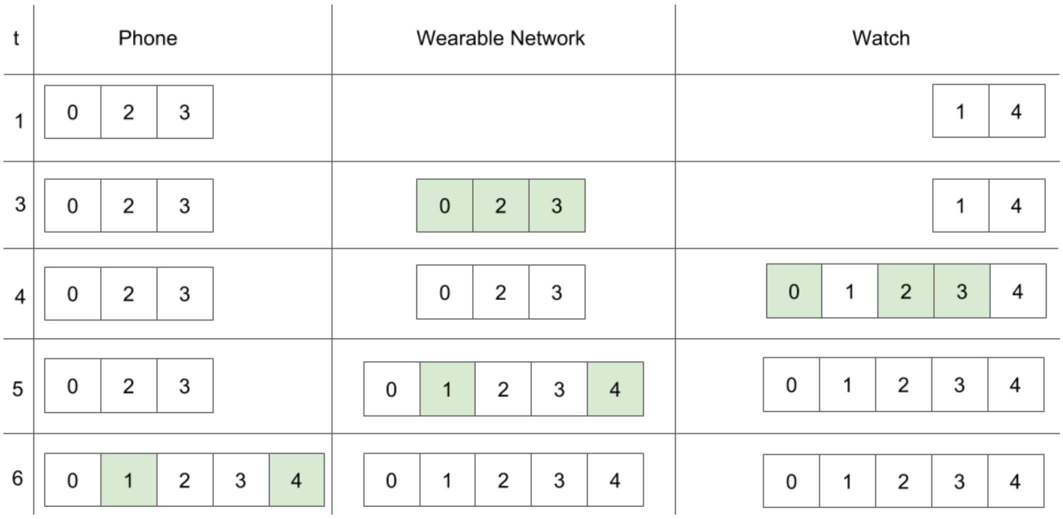

Figure 4.7: Conceptual diagram of the synchronization of activities between phone and watch

While the implemented synchronization algorithm has its inefficiencies, it has advantages that are more important for this project: The mechanism is very solid. Activities that were tracked on one device will always be pushed to the other. This also makes it possible to, e.g., deinstall the phone application without losing data if the watch application is not deleted. The algorithm will synchronize all activities that are present on the watch app to the phone app. Because the synchronization mechanism only gets executed when a new activity is added (which is usually less than ten times a day) efficiency was deprioritized in favor of stability.

When an activity gets deleted on the phone a very similar mechanism is executed to synchronize the deleted activities to the watch. The watch then deletes the concerning activities from its database. Editing of activities is implemented as deleting of the old activity and adding a new one.

Heart Rates

As mentioned previously, the sending of collected heart rates from the watch to the phone is implemented differently to the synchronization of the other data types. The reason for this is the larger amount of data that is generated by the heart rate measurement compared to other data types. An approach that allows the deletion of HR data from the watch, after it has been sent is also preferable. With the aforementioned algorithm, this is not possible because it will re-synchronize the deleted data. The sending of HR data is performed by a combination of the DataAPI and the MessageAPI. The communication of heart rates from the watch to the phone can be seen in the UML sequence diagram in Figure 4.8. The depicted situation in Figure 4.8 shows the sending of a large number of entries of the **Heartrate** table from the watch to the phone. The buffered number of unsynchronized heart rates can become very large, depending on the synchronization interval and the measurement interval. Because the DataAPI only allows a node to put less than 100 kilobytes of data in the Wearable Network, the collected heart rates have to be split into packages if they exceed that size. In this example, there were more than 100 kilobytes of heart rate data collected and not synchronized.

The sending of the heart rates begins with the execution of the **HeartRateSyncJobService**, which is started at user-defined intervals. Now, all unsynchronized entries are gathered from the **Heartrate** database. After this, the entries are split into packages of 3000 or fewer entries. The packages and their contents are ordered by the timestamp of the measurements. The splitting ensures that no package will exceed 100 kilobytes. Subsequently, the packages are sent using the DataAPI via the Wearable Network. After the phone receives each package, it inserts the received timestamps and heart rates into its local database. Then, it sends a message with the MessageAPI to the watch on the *CONFIRM_PATH*. The message's payload contains the first and the last timestamp of the package's entries. After the watch received this message, it deletes all measurements between the received timestamps to save storage on the watch. The phone now starts to calculate the computed stress metric and synchronizes it back to the watch.

The confirm message enables a reliable synchronization: It is not possible for the watch to delete heart rate data before it was sent to the phone and securely received. A package may have to be synchronized twice because the confirmation message is not delivered via the network. In this case, the phone's database handler will recognize that the received data is already in the database and ignore it.

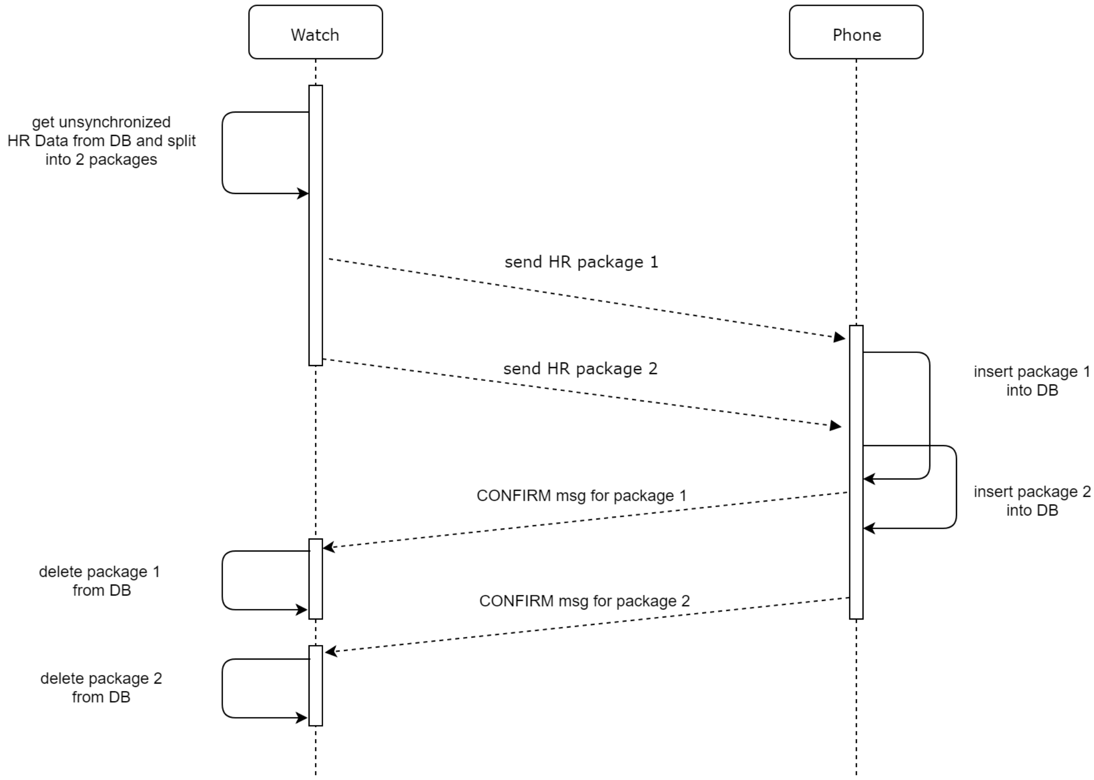

Figure 4.8: UML sequence diagram showing the communication of heart rates from the watch to the phone

4.5.3 Messages

As mentioned earlier, Stila uses the MessageAPI to realize auxiliary communication. The classes **MessageToWatchUtil** and **MessageToPhoneUtil** can be used to send a message to the other device. The **DataLayerListenerServices** on the receiving device further processes the message depending on its path and payload. To make the messaging code easily reusable, Java enumerations were created that define the purpose of the message. These are called **MessageMode** and can be found in the watch and the phone module. The supported message modes for each device are explained in the following:

- **Phone**
  - ***RESYNC:*** This message is sent by the phone after a user presses the synchronization button in the phone application. It triggers an instant synchronization of heart rates from the watch to the phone, independent of synchronization intervals.
  - **CONFIRM_DELIVERY:** This message mode confirms the delivery of heart rates by the phone. Its payload contains the first and the last timestamp of the received heart rates.
  - **CHECK_ONLINE:** Messages of this mode are used to check the connection between phone and watch. It allows informative error messages by telling the user whether a connection to the watch could be established.

- **Watch**
  - **CONFIRM_ONLINE:** This message mode is the counterpart of the CHECK_ONLINE message. It is sent by the watch as a response to the phone. It confirms that there is a connection between the two devices.

4.6 Stress Computation

Stila uses computed stress to give the user feedback about her stress levels. Computed stress is defined as “stress computationally derived from instantaneous measures of stress symptoms obtained by non-invasive methods” [50]. The Stila applications use heart rate measurements as a non-invasive method. In order to derive the stress intensity from the heart rate, the heart rate variability (HRV) has to be calculated. HRV is an indicator of a person’s dynamic and cumulative load. This makes HRV a good heuristic for the stress a person feels [50]. A low HRV can be interpreted as a high level of stress, while a high HRV suggest a low level of stress. The following section describes how the Stila phone application calculates the heart rate variability from heart rate measurements gathered by the watch:

4.6.1 Formulas

Stila derives the HRV score from the root mean square of the successive differences (RMSSD) in the heart rate. This metric is dependent on the intervals between consecutive heartbeats (RR)[21]. The following equations show the derivation from heart rates to HRV. The RR can be calculated by dividing 60 by the measured heart rates per minute (HR). This can be seen in Equation 4.1. Equation 4.2 calculates the RMSSD on a specific window of HR measurements [4]. In the Stila application, this window is 10 minutes long and contains all measured heart rates in this timespan. The HRV score for a time window is subsequently calculated from the RMSSD as shown in equation 4.3 [21].

$$RR = 60/HR \qquad , HR := measured\ heart\ rate\ in\ beats\ per\ minute \tag{4.1}$$

$$RMSSD = \sqrt{\frac{\sum_{i=1}^{n}(RR_i - RR_{i-1})^2}{n-1}} \qquad , n := number\ of\ HR\ measurements \tag{4.2}$$

$$HRV = ln(RMSSD \times 1000) \times 20 \tag{4.3}$$

As mentioned before, a high HRV points to a low stress level. This can be counterintuitive to users. To make the HRV score easier to understand, Marcel Heil [21] suggested the introduction of a computed stress metric by inverting the HRV score. This is done by subtracting a large enough constant. Because HRV scores usually range from 50 to 80, he used the following formula for calculating the computed stress metric:

$$Stress_{computed} = 110 - HRV \tag{4.4}$$

The result of this calculation is used in Stila to present the stress graphs, estimate the user's stress level in the smart notifications and to power the stress complications.

4.6.2 Implementation in Stila

Before the start of this thesis, there was already an implementation of the aforementioned formulas by Marcel Heil [21]. This implementation was interwoven with the usage of the Fitbit API and could not be used with Wear OS. To counteract this problem, the stress computation was completely rewritten using a more modular approach. To save battery power, the somewhat costly Stress computation is being done exclusively on the phone and then communicated back to the watch. For this reason, the **WearHeartrateAdapter** class was added to the phone application. The class bundles the functionality needed to save communicated heart rates and calculate HRV and computed stress values. Despite its name, the WearHeartrateAdapter can be used with an arbitrary heart rate monitor. This enables an expansion of supported devices in the future.

Figure 4.9 shows a UML activity diagram that presents the logic behind the processing of received heart rates and the calculation of HRV and the computed stress. For brevity and because the metrics are linearly dependent, the HRV score and computed stress level are denoted solely as HRV in the following. The calculation of the computed stress metric happens in the same step as the calculation of the HRV score.

After the phone received new heart rates from the watch, the *processNewHRData* method is called in the WearHeartrateAdapter class. The heart rates and the timestamps are passed as parameters. The method starts the processing by saving the heart rates and their timestamps as entries in the **Heartrate** database of the phone. Because the algorithm works with 10-minute windows, it does not make sense to execute the calculation more often than that: If the last HRV calculation was less than 10 minutes ago, the method returns and waits for the next heart rates. If the last calculation was more than 10 minutes ago, the algorithm gathers all heart rate data from the last HRV calculation up to the current time.

These data are subsequently searched for distinct 10-minute windows. This step is necessary because it is not known how much data the watch sent to the phone. Depending on the synchronization interval, it could have received more than 10 minutes of heart rate data. After all 10-minutes windows are identified, it is checked whether they contain enough data. In the beta version of the Stila application, a window is deemed valid if it contained more than 3 data points. It might be advisable to increase this value in further studies, especially because smart watches are usually capable of measuring the heart rate every second.

If the windows are valid, the HRV score is calculated with the aforementioned formula. After that the computed stress metric is derived from the HRV score. The HRV entry's timestamp is generated by rounding the first timestamp of the window down to the next full 10-minute timestamp. E.g., when a window ranging from 13:42 - 13:52 was processed, the calculated HRV entry gets saved with the timestamp 13:40.

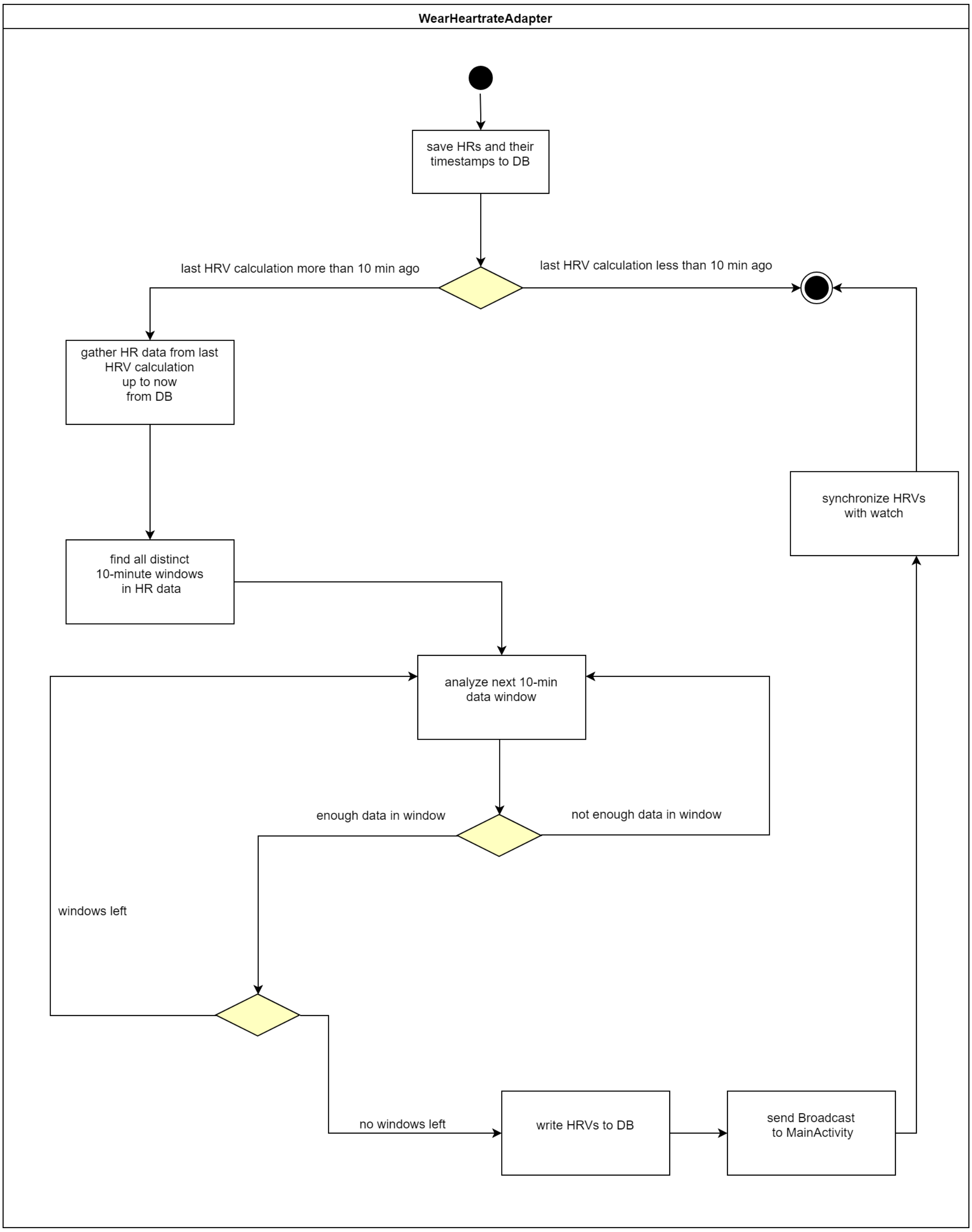

Figure 4.9: UML activity diagram showing the logic behind the calculation of HRV and the computed stress level on the phone

After all windows have been processed, the computed stress values together with the HRV score and the corresponding timestamp are saved in the HRV database of the phone. Subsequently, a system-wide broadcast is sent to inform the stress graphs in the MainActivity of the application to refresh because new data is available. Finally, the newly generated HRV database entries are synchronized to the watch.

The design of this algorithm is very versatile. It allows the HR to be streamed to the phone in real time, while simultaneously also allowing HRV calculation from buffered data.

4.6.3 Stress Indicators

In order to facilitate stress complications, smart notifications and stress graphs, the Stila system has to be able to decide whether a user is "stressed", "neutral" or "relaxed". This makes it necessary to have an algorithm which can categorize a computed stress value in one of these classes.

For the limited scope of this study, a straightforward stress indicator algorithm was built, which decides the category of the computed stress value on the basis of fixed bounds. The class **FixedBoundsStressIndicator** on the Wear OS application categorizes a computed stress value as 'relaxed' when it is below 10. Values higher than 10 and lower than or equal to 30 are 'neutral'. A computed stress value higher than 30 is marked as "stressed". Early tests have shown that very short intervals between measurements tend to produce far higher computed stress values because the variability in the heart rate can be calculated more accurately. To account for this, the FixedBoundsStressIndicator multiplies the bounds by the *MEASURE_FACTOR*. If the watch measures the heart rate continuously, the *MEASURE_FACTOR* is set to 2.0, else it is set to 1.0. Without this multiplication, the watch would always indicate that the user is stressed.

As mentioned before, the used stress indicator is rather simple in its logic. Future work could include a "smart" stress indicator that calculates the bounds for the stress categories with the use of historical data dynamically. That would also enable personalized stress indicators for every user. To make such additions easily possible the FixedBoundsStressIndicator implements the **StressedIndicatorAlgorithm** interface. Better stress indicators can easily be integrated into the Stila application by implementing this interface.

4.7 Interface Design and User Experience

This section describes the realization of the interface design and other elements that improve the user experience. The most relevant concepts in relation to smart watches are described.

4.7.1 Watch Application Interface

The interface of the Stila watch application has to cater to two different form factors: Square and round watches. Wear OS applications are expected to support both display types. Therefore the Stila watch application has been developed to be usable on round and square watches. This includes the Stila watch application and the watch face. Figure 4.10 shows the Activity Overview on a square (left) and a round (right) display. To facilitate this adaptability, the **Wearable UI library** can be used. The Wearable UI library is part of the *android.support* library and includes several components that ease the development of user interfaces for Wear OS [37].

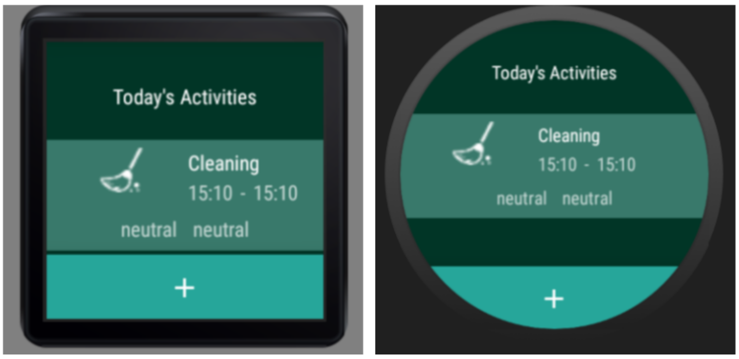

Figure 4.10: Activity overview on a square (left) and a round display (right)

The Stila Wear OS application uses several components of the Wearable UI library, including **WearableRecylcerViews** that allow the display of scrollable lists on round watches. Another important usage of the Wearable UI library is the **BoxInsetLayout**. This root layout can be used to make sure that a displayed item is visible on both round and square displays without scrolling on either of the devices. The library defines a box in the middle of the watch's screen. The box's dimensions are small enough to be displayed on a round device without scrolling or cropping. Other elements can be positioned inside the box in relation to its borders. Figure 4.11 shows a round watch with an exemplary BoxInsetLayout. The BoxInsetLayout is colored in a lighter gray than its background. This shows, how the limited space on the screen has to be further minimized in order to support square and round displays.

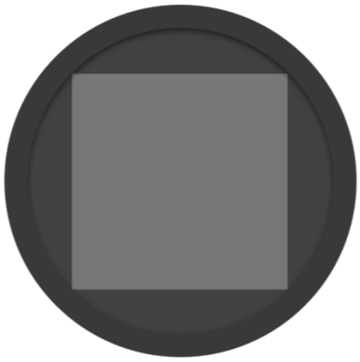

Figure 4.11: Example of the BoxInsetLayout on a round display

The navigation and action drawers of the app are also implemented with the help of the Wearable UI library. To add the drawers, the layout of the activity has to wrapped in a **WearableDrawerLayout** and contain the **WearableNavigationDrawerView** and the **WearableActionDrawerView**. The menus have to be populated dynamically in the activity's Java code.

4.7.2 Onboardings and Tutorials

An important part of the user's experience is its start: The onboarding process. The user is greeted with a set of tutorials on the watch and the phone. This section describes how they were implemented.

Onboarding Tunnel on the Phone

This tunnel was built using the *Ahoy! Onboarding*[2] library for Android. Stila's **IntroductionOnboardingActivity** class in the app module was created to implement the library. With the help of this class, the onboarding tunnel can easily be changed or extended in the future.

Usage Tutorial Tunnel on the Phone

This tunnel explains the app's most important features by darkening the screen and highlighting certain positions while showing tutorial texts. The feature was implemented by utilizing the *Spotlight*[3] Android library. The class **TutorialOverlayGenerator** was added to the app module to allow the easy creation of further tutorials within the application.

Onboarding Tunnel on the Watch

The onboarding tunnel on the watch has a similar purpose to the onboarding tunnel on the phone. Because of the different form factors, the same library could not be used. Furthermore, no other Android libraries were available for the creation of onboarding dialogs on Wear OS devices. Therefore, a solution was developed which is based on a **ViewPager**. A ViewPager can be used to control how the user navigates between fragments in an Android activity. This was utilized to create tutorial cards as fragments that can be traversed by swiping from the bottom of the screen to the top. After the user scrolled through the whole tutorial, she gets redirected to the Wear OS watch face picker. This is implemented by the **PickWatchfaceOnboardingFragment**, which calls an implicit Intent on the watch's WallpaperManager. The class **OnboardingGridViewPager** in the wear module can be used to add further cards to the tutorial.

[2] https://github.com/codemybrainsout/ahoy-onboarding
[3] https://github.com/TakuSemba/Spotlight

CHAPTER 5

Evaluation

5.1 Hypotheses

The goal of this thesis is to increase the user engagement in the Stila application so as to foster stress awareness. This goal is to be achieved by improving the application with approaches from the fields of **Behavior Change Theory**, **Persuasive Technology** and **Wearables**. To make the achievement of the goal verifiable, it is split into three concrete hypotheses:

Hypothesis 1:
Users of the improved Stila application track their activities more often than users of the unimproved Stila application.

Hypothesis 2:
The changes from the unimproved to the improved Stila application increase the ability of its users to track their activities.

Hypothesis 3:
The improved Stila application increases the subjective stress awareness of its users compared with the unimproved version.

5.2 Study Design

A study was conducted to verify the aforementioned hypotheses. The following section describes the design of the study.

5.2.1 Overview

The Stila application was evaluated with the help of a user study. The study participants were split into two groups: The **Fitbit** and the **Wear OS** group. The Fitbit group used the unimproved Stila application in Fitbit Mode, without its persuasive features. Participants of the Wear OS group activated the Wear OS mode, which gave them access to the persuasive features of the improved Stila application as described in this thesis: This includes the watch application, the onboarding tunnels, stress complications and so on. In other words, the ability dimension of the Wear OS group was manipulated, while the Fitbit group's was not.

To make the results comparable against the backdrop of Fogg's Behavior Model both application versions needed the introduction of triggers. The Wear OS application's persuasive capabilities allowed the introduction of smart notifications as triggers. The Fitbit application was enriched by an ordinary notification which triggers the user every three hours (see chapter 3.4 *Triggers*). The addition of triggers ensures that the manipulation of the ability dimension is the only independent variable.

The participants of the study were asked to use the Stila application for 15 days (17th July - 31st July 2018). During this period of time the groups' usage of the application was tracked using software usage analysis tools. After the study, a questionnaire was sent to the participants, which explored the users' personal experience with the applications. The study was conducted online. Communication with participants was performed via email.

5.2.2 Participants

Characteristics of the Sample

86 subjects participated in the study. The participants were equally distributed between the Wear OS group and the Fitbit group. Most of the subjects used their own devices to participate in the study (see *Recruitment of Participants*). The subjects were between **18** and **61** years old at the time of the study. The average age was **31.52** years. Participants in the Wear OS group predominantly (**83.7%**) identified as male. The Fitbit group had slightly more (**51.2%**) female subjects. The age and gender structure of both groups can be seen in Figure 5.1.

	Wear OS	**Fitbit**	**Total**
Number	43	43	86
Gender	male: 83.7% female: 16.3%	male: 48.8% female: 51.2%	male: 66.3% female: 33.7%
Age	median: 30 mean: 30.23	median: 30 mean: 32.86	median: 30 mean: 31.52

Table 5.1: Age and gender structure of the sample

People from **19** different countries participated in this study. The full list of countries of origin in alphabetical order is: *Australia, Belgium, Canada, Czechia, Denmark, France, Germany, Ireland, Israel, Italy, Japan, Netherlands, Poland, Romania, Singapore, Slovenia, Switzerland, United Kingdom,* and *United States of America*

15.1% (Wear OS: **9.3%,** Fitbit: **20.9%**) of the participants indicated that they suffer from stress-related ailments and were therefore interested in the study. Other reasons for participation included general interest in the measured data or the hope to identify hidden stressors.

68.8% of all subjects were working a full-time job in the period of the study. The other participants were either studying, traveling or enjoyed free time.

Recruitment of Participants

The participants were recruited through different channels. The majority (**84.9%**) of subject were found in the online community *Reddit*.[1] **2.3%** of participants were recruited through the *Quantified Self Forums*.[2] The rest was found in the university environment via newsletters, in lectures, and in Facebook groups. Students of Ludwig-Maximilian University of Munich were able to borrow Wear OS and Fitbit devices. This offer was used by nine participants (**9.6%**). Because these participants had no previous experience with the devices, the smart watches and Fitbits were handed over to the participants four days before the start of the study. The nine university-affiliated participants were encouraged to get to know the devices by trying them out and using them as they pleased before the study started. This was done to compensate for possible novelty effects that could artificially increase engagement in users of borrowed devices.

Participation in the study was on a voluntary basis and no inducements to participate were offered. At the beginning of the study, the subjects received an email with a link to the application and the request to use it on a daily basis. Participants were encouraged to track their activities as often as possible. Furthermore, a mailbox was created which could be used to report technical errors in the application. Both groups received the same information about the study. Participants were not told that they were part of an A/B test or that the goal of the study was to prove the superiority of the Wear OS application. After the the study ended, the participants received another email with the link to an online questionnaire.

5.2.3 Data Processing

The data collected by the study was processed using the data visualization software *Tableau*. The significance of the data presented in the next sections is evaluated based on p-values. The p-values were calculated using **one-tailed Student's t-tests**. These p-values show the probability whether the tested samples come from the same underlying population. Because the study aims to show that the additions to the Stila application created changes in the user engagement, low p-values are desirable. The sample size was not determined with the use of a power analysis but emerged organically. Consequently the p-values can not be used to verify hypotheses based on a fixed confidence level. Nevertheless the p-values can inform about the strength of measured effects.

5.3 Usage Analysis

5.3.1 Implementation

Within the time of the study, the Stila applications were monitored using the open source tool *Matomo*.[3] The Matomo Android SDK[4] was used to collect information about the users'

[1] https://www.reddit.com
[2] https://forum.quantifiedself.com
[3] https://matomo.org
[4] https://github.com/matomo-org/piwik-sdk-android

interaction with the Stila applications. While Matomo is generally aims at the usage as a web analytics platform, it can be used in Android applications with the help of custom-created **events**. The recording of an event on a Matomo server can be executed in the Android code. When a user triggers an event by performing a certain interaction, the Matomo server is informed by the Android device. Both the Stila Android and the Wear OS application have been extended to send interaction information to the Matomo server.

The users were identified by their Google ID. This allows the aggregation of the interactions of one user on different devices. A participant who uses the phone and the watch app is therefore not counted as two users. Each tracked event was enriched by the information whether the user was part of the Wear OS or the Fitbit group. This allowed the separation of the two groups in the collected data.

The following data was collected in the form of Matomo events:

- How often was each screen visited?
- How many activities were tracked on which device?
- How many notifications were sent?
- How many notifications resulted in the tracking of an activity?
- How often was the watch face style changed?
- How often was the measurement interval changed?
- How many visits resulted in the tracking of an activity?
- How much time elapsed between the opening of the Stila application and the tracking of an activity?

5.3.2 Results

Number of Users

A discrepancy between the registered participants of the study and the actual unique users of the application was observed. As mentioned above, both groups had **43** registered participants. However, the collected data only showed **36** Wear OS and **41** Fitbit users during the study. This means that several subjects registered for the study but subsequently did not download the Stila application when the study started. Furthermore, the number of users steadily decreased during the study. It was observed that the user numbers fell at roughly the same rate for both groups. The loss of users was especially great in the first six days of the study. The user numbers stabilized towards the end of the study at approximately **10** users per each group. Figure 5.1 shows the evolution of unique daily users over the time of the study. The x-axis shows the days of July 2018 during which the study took place. Furthermore, it can be seen that there were never more than **62** active users on a single day.

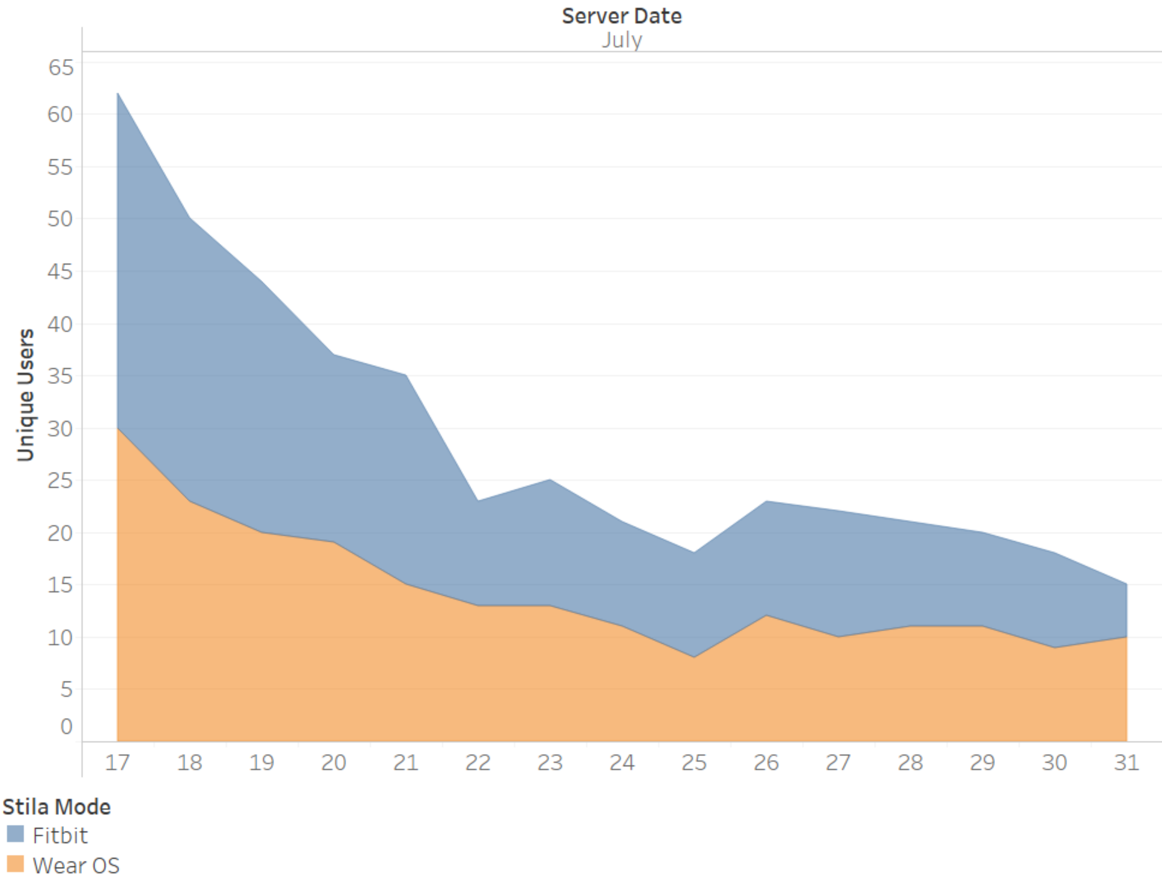

Figure 5.1: Evolution of unique users over time

Tracked Activities

Wear OS users tracked their activities more often than Fitbit users. Figure 5.2 shows the absolute number of tracked activities for both groups. In total **1203** activities were tracked. Wear OS users tracked **657** activities while participants of the Fitbit group recorded only **546**. This means that the Wear OS group tracked **20.33%** more activities than the Fitbit group.

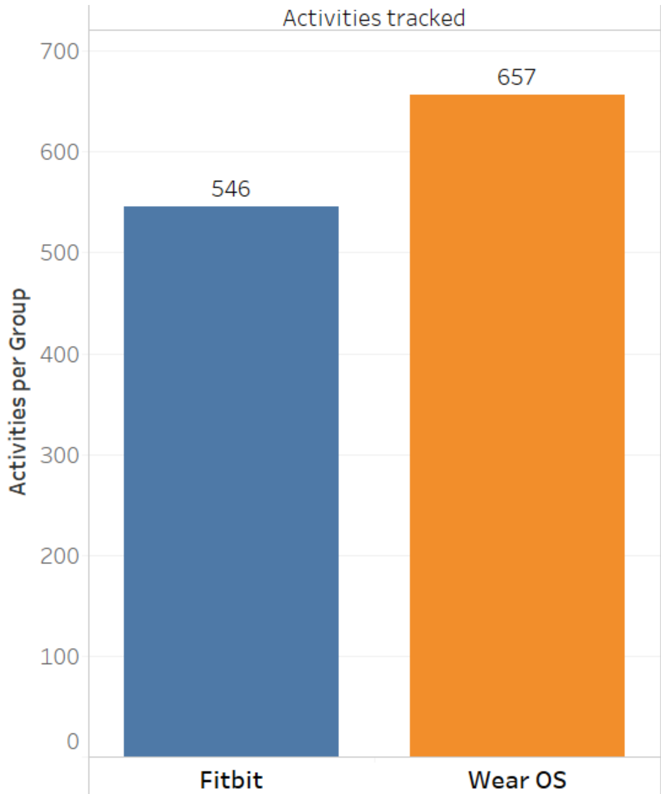

Figure 5.2: Absolute number of tracked activities for each group

As mentioned above, it was also analyzed on which device (phone or watch) the Wear OS group tracked their activities. Figure 5.3 shows that users preferred the watch (**496** activities) over the phone (**161** activities).

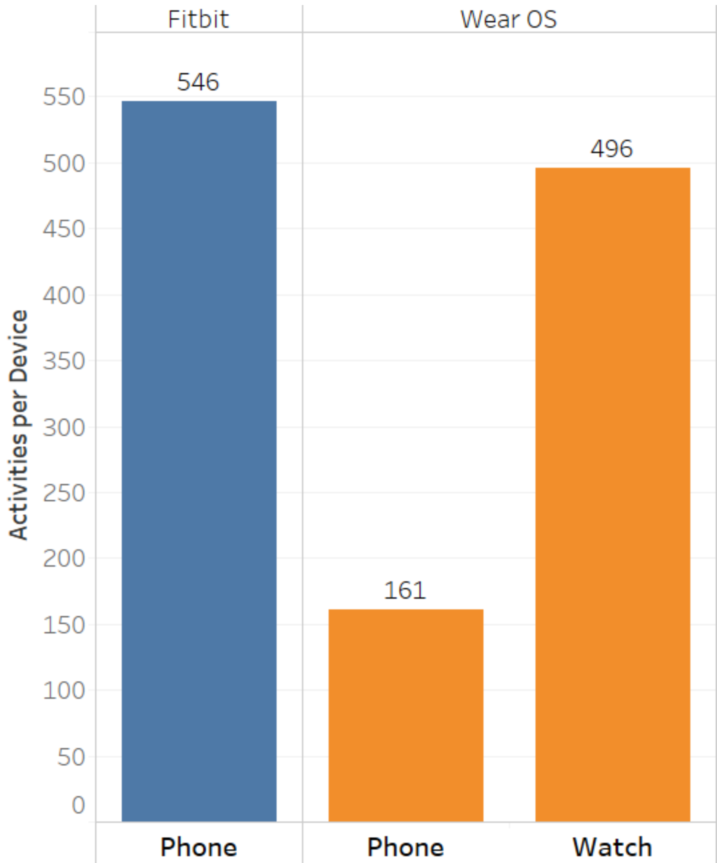

Figure 5.3: Absolute number of tracked activities per device

Because the actual number of participants differed between the groups, the mean number of tracked activities per group was calculated. Only users who tracked at least one activity were used as basis for this calculation (Wear OS: 26 users, Fitbit: 33 users). As can be seen in Figure 5.4, users of the Wear OS group tracked a mean of **25.27** activities while users of the Fitbit group only recorded a mean of **16.55** activities. This signifies that the average Wear OS user tracked **8.72** activities more, which corresponds to an increase of **52.69%** ($p = 0.1654$).

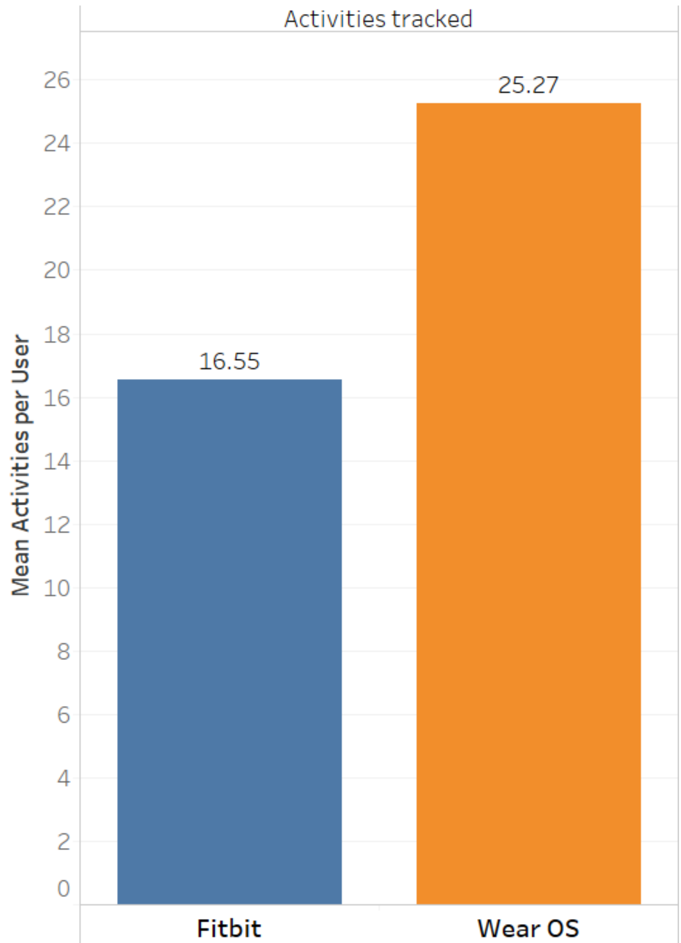

Figure 5.4: Mean number of tracked activities per user

Figure 5.5 shows the temporal development of the average number of tracked activities per daily active user. The x-axis again shows the days of July 2018, when the study was conducted. The bars are split into Wear OS and Fitbit users. It can be seen that on 9 of the 15 days an average Wear OS user tracked more activities than an average Fitbit user.

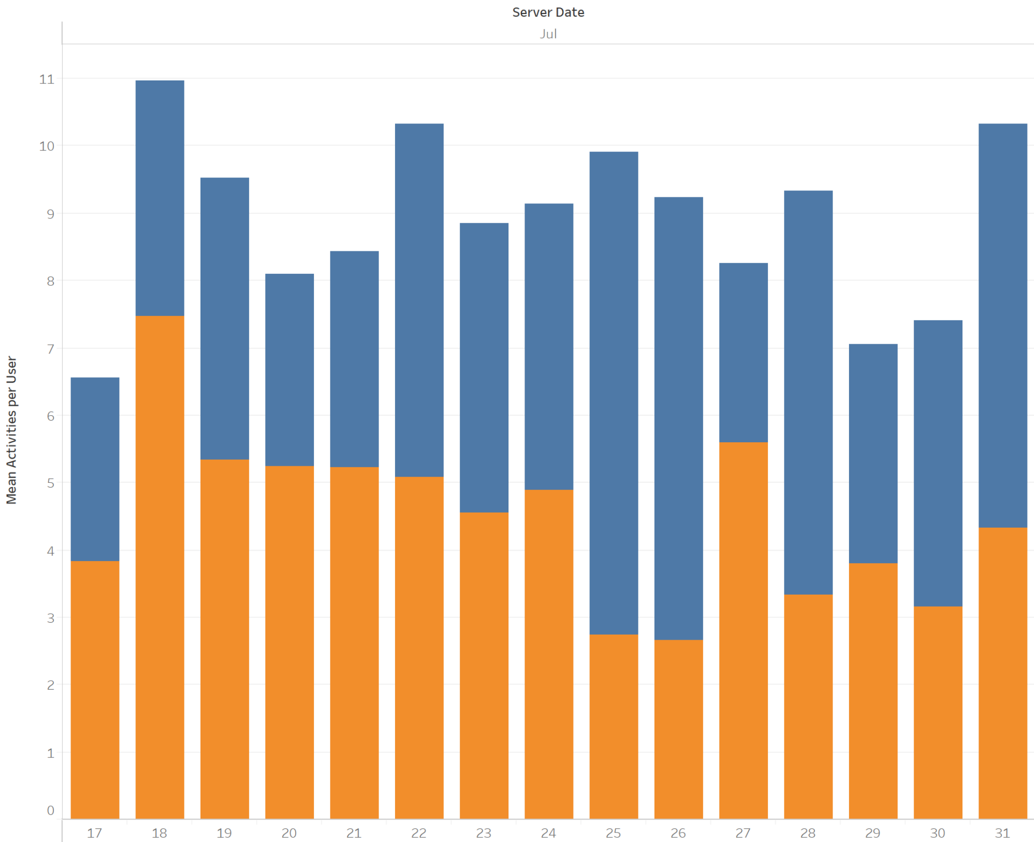

Figure 5.5: Temporal development of the mean number of tracked activities per daily active user

When looking at the absolute numbers of tracked activities per day in Figure 5.6, it can be seen that the Fitbit group only tracked more activities than the Wear OS group on three days. Especially at the beginning of the study, the Wear OS users recorded more activities than the Fitbit users.

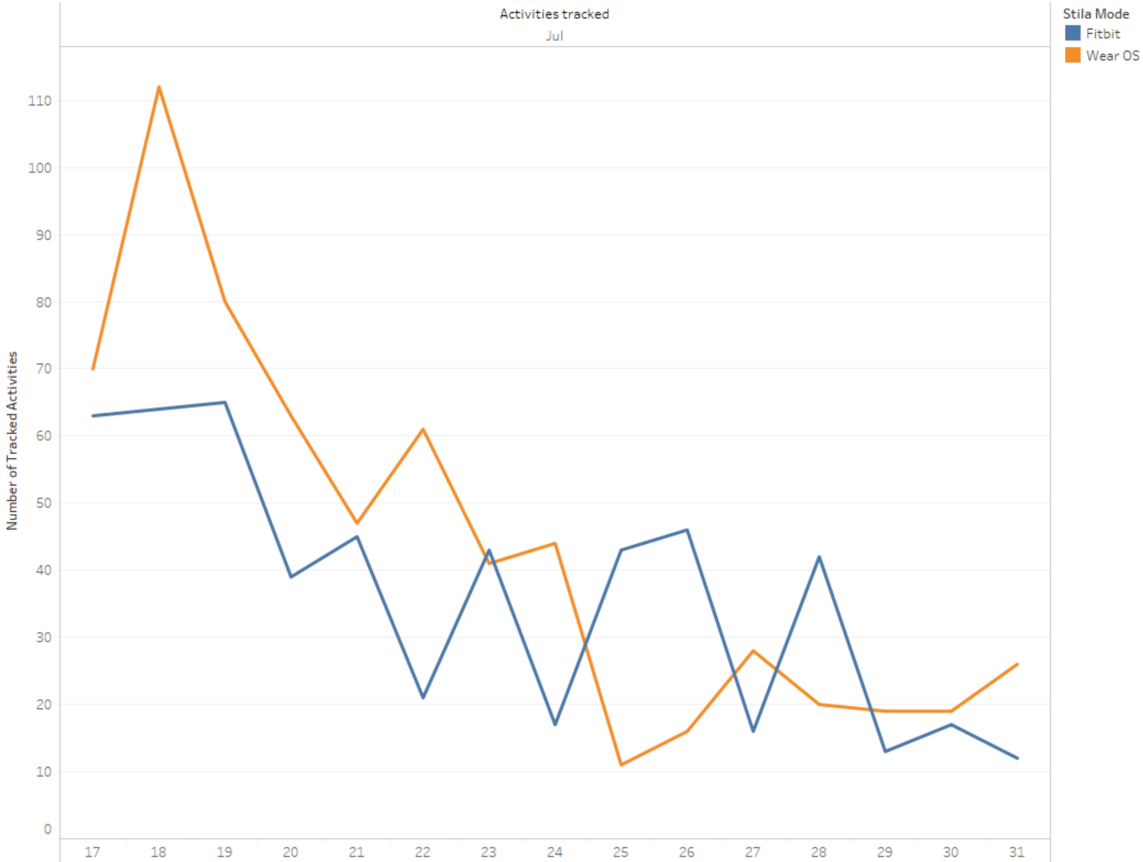

Figure 5.6: Tracked activities per day

Conversion Rate

In the context of this thesis the term *Conversion Rate* denotes the percentage of visits to the application that resulted in the target behavior (the recording of an activity). The conversion rate can provide details about the reasons why the application was opened. Wear OS users visited the Stila applications a total of **1576** times. **650** visits resulted in the recording of an activity. This corresponds to a conversion rate of **41.24%**. Fitbit users, on the other hand, opened the application **583** times, recording an activity in **546** cases. This is a conversion rate of **93.65%**. This means, the Wear OS group opened the Stila applications **170.33%** more often, but had a far lower conversion rate. In the Fitbit group, almost every visit of the application resulted in the target behavior. When comparing these numbers with the absolute number of tracked activities (see Figure 5.2) one can also see that most users only tracked **1** activity per visit. The change in the conversion rate is significant beyond the 99% level of confidence.

Triggers

As mentioned before, Triggers were implemented as push notifications on the watch (Wear OS group) and the phone (Fitbit group). The Wear OS group was only notified when heuristics concluded that it was an opportune time, while Fitbit users were notified when they did not open the app for three hours. It was counted how often notifications were sent to the user and whether the notification resulted in the tracking of an activity. If yes, this trigger is called *successful*. The Wear OS application sent **538** notifications to its users. **9.85%** of these were successful. The Fitbit users received **711** notification of which **7.88%** resulted in the recording of an activity. This corresponds to an increase of **2.74%** (p = 0.1137).

Tailoring

Wear OS user could personalize their experience by changing the color of their watch face. This was used by **16 (44.44%)** Wear OS users. **18 (50%)** users in Wear OS mode changed the measurement interval in the settings of the watch application. **10 (27.77%)** participants used both personalization features.

13 (36.11%) Wear OS users created custom activity types. These users tracked **31.17 activities** on average over the course of the study. Users who did not create custom activity types recorded only **20.21 activities** on average. This means that users who created custom activities averagely created **10.96** (p = 0.2893) activities more than users who did not.

The complications on the Stila watch faces were changed by **18 (50%)** Wear OS users. On average there were more (p = 0.0200) activities recorded by participants who changed their watch face (**33.17** activities) than by participants who did not (**8.42** activities).

Watch Faces

Wear OS users could freely decide which watch face they wanted to use. **27 (75%)** picked the digital watch face while **9 (25%)** chose the analog watch face. This shows that the design digital watch face was more popular in the sample.

Time

The time needed to fill out the different dimensions of an activity was also measured. Wear OS users needed **19 seconds** on average to track their feelings for an activity on the watch. Fitbit users needed **16 seconds** on average to fill out all dimensions. Consequently, participants needed **18.75% longer** to track their activities on the watch than they needed on the phone. This difference is statistically significant beyond the 99% level of confidence. Not included is the access time to the device and the application, which is not measurable with the help of analytic tools.

5.4 User Survey

5.4.1 Implementation

After the user study ended, an online survey was sent to the participants. The survey explored the subjective experiences the participants had with the applications. Two separate questionnaires were sent to the groups. The Wear OS questionnaire contained several extra questions which compared the users' experiences with the watch application to that with the phone application. The Wear OS questionnaire can be found in the appendix of this thesis.

24 (Wear OS: **11**, Fitbit: **13**) participants filled out the survey. Due to the small sample size, the questionnaire has less statistical significance. Furthermore, the gender structure of the survey sample differed from the gender structure of the user study: Fitbit participants of the user study were 51.2% female, while the survey respondents were 76.2% female. The survey respondents of the Wear OS group had roughly the same gender mix as in the user study. In sum, proportionally more women filled out the survey than participated in the study. Side effects on the data cannot be ruled out. Nevertheless, the collected data offers interesting insight and clues about the groups' experiences.

5.4.2 Results

The groups were asked to rate multiple statements on a Likert scale from 1 to 5. In the used symmetric agree-disagree scale, 1 meant a strong disagreement with the statement while 5 meant a strong agreement with the statement. Furthermore, the participants were asked several multiple and single choice questions. In some cases, the participants could respond with free text comments.

The following sections analyze the participants' answers to the survey:

Understandability and Learnability

The participants were asked to rate the statement "I fully understood the purpose of the Stila App." The average score given on the aforementioned Likert scale for Wear OS users was **4.30**. Fitbit users averagely rated this statement with **4.00**. This shows that both groups had a rather good understanding of Stila's purpose, with the Wear OS group being slightly better (p = 0.1929).

Furthermore, the participants were asked when they understood the purpose of the application. Figure 5.7 shows the answers of the participants by group. It can be seen that **54.5%** of the Wear OS group understood the application right away, while only **46.2%** of the Fitbit group did so (p = 0.3488). Neither group had participants who never understood the purpose.

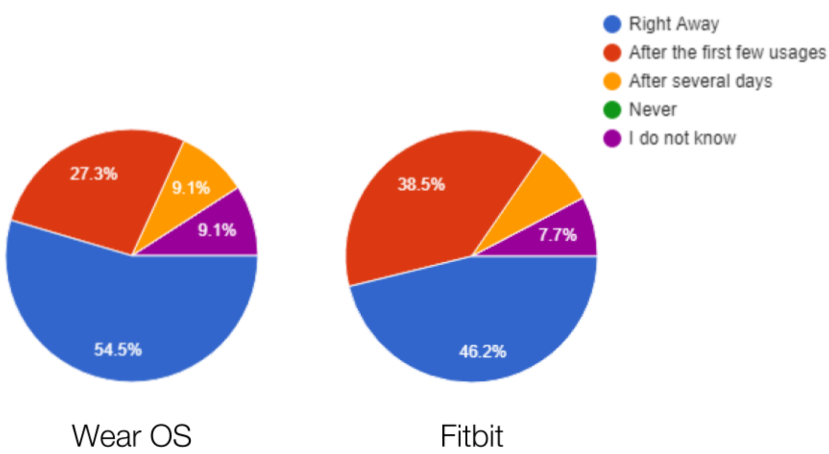

Figure 5.7: Answers to the question "When did you understand the purpose of the Stila App?"

Participants were also asked to rate the statement "Stila was easy to learn". This statement was averagely rated **4.36** in the Wear OS group and **3.77** in the Fitbit group. This means that the Wear OS group rated the overall learnability of the applications slightly better than the Fitbit group (p = 0.0454).

Motivation

Regarding their motivation, the participants rated the statement "My motivation to use the Stila App at the beginning of the study was high". The Wear OS users confirmed this with an average score of **4.64** on the Likert scale. Participants of the Fitbit group agreed to this statement with an average score of **4.54**. This means that participants of both groups were very motivated to use the Stila applications at the beginning of the study. The difference between the group has a low statistic significance of p = 0.3670.

The users were also asked how their motivation changed during the study. The Fitbit group rated the statement "My motivation to use the App decreased during the study" with **3.31**. The Wear OS group rated the statement with **3.09**. This question that explored the motivation of the users was again answered very similar by the two groups (p = 0.3341).

The participants that stopped to use the app before the study ended were asked to specify why they did so. Most Fitbit users **(87.5%)** stated, that they simply forgot to use the application. Only three participants of the Wear OS group answered this question, which did not lead to an evaluable result.

Participants were also asked, whether they planned to continue their usage of the Stila applications. **15.4%** of the Fitbit group and **9.1%** of the Wear OS group ruled this out by answering "No" (p = 0.3267). Figure 5.8 shows the answers that were given by the groups as pie charts.

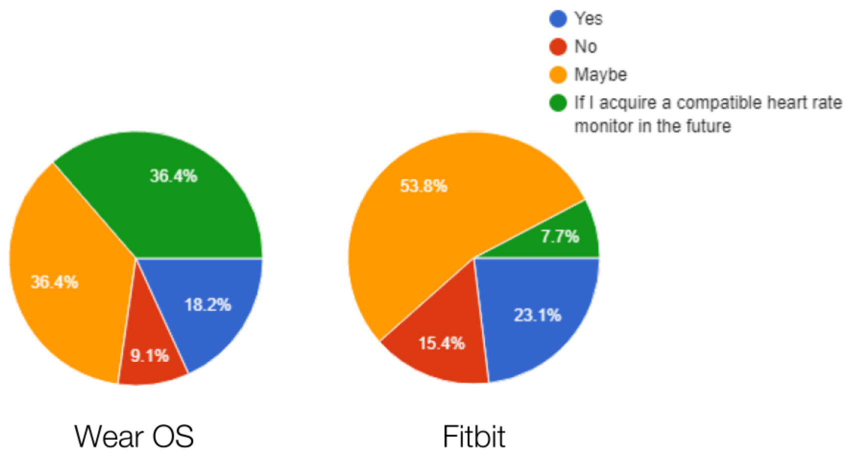

Figure 5.8: Answers to the question "Will you continue your usage of the Stila Apps?"

Usability and Simplicity

The participants were asked to rate the statement "The important features are quickly accessible". Wear OS users rated this statement better by giving an average score of **4.46**. Fitbit users only rated this statement with a score of **3.54** (p = 0.0119).

Users were also asked, whether the Stila applications were easy to use. Wear OS users were asked two distinct questions for the phone and the watch, Fitbit users were only asked about the phone application. In sum, the watch application was rated better with an average score of **4.36** over the phone application that had an average score of **3.67**. This shows that the Stila watch application was voted considerably simpler to use than the phone application (p = 0.0182).

Wear OS users (who experienced both the watch and the phone application) were asked to rate, whether they thought it was more convenient to track their activities on the phone or the watch. The results show that **72.7%** of Wear OS users found it more convenient to use the watch application for this task. Only **18.2%** preferred the activity tracking via the phone. Figure 5.9 shows the answers to this question as a pie chart.

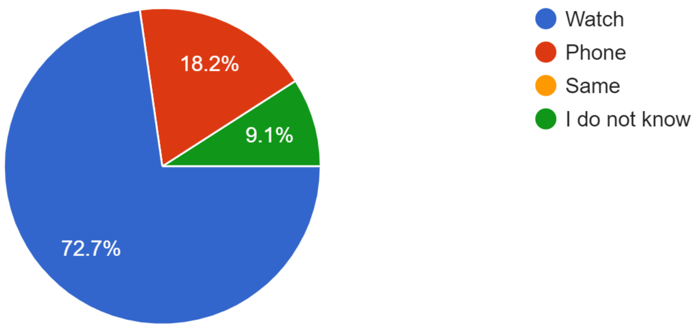

Figure 5.9: Answers to the question "Was it more convenient to record your activities on the watch or on the phone?"

The participants also had the possibility to explain why they preferred a device for the recording of activities. Most participants cited the easier access to the watch as the main reason for their decision. One participant stated that they liked the stop-watch feature allowing intuitive time tracking in the Wear OS application. Another participant indicated that they preferred the watch for recording current activities but used the phone for the tracking of past activities: "It was easier to record on the watch current activities. Past activities were actually better added from the phone. As for why? On the watch you tap the complication, choose and start. The watch is always with me, that is why I'm using a smart watch." The subjects that preferred the phone over the watch, mentioned technical problems with the watch and the bigger screen of the phone as reasons for their conclusion.

Furthermore, the users of the Wear OS application were asked to rate the usability of the application. They confirmed that the application was easy to navigate (**4.09**), that they liked the design of the watch application (**4.55**) and the watch face (**4.00**) and that the buttons and texts were big enough (**4.73**). Overall, this speaks for a high usability.

User Engagement

The participants of the Wear OS and the Fitbit group were asked how often they interacted with the phone and the watch application. The Wear OS group was again asked to protocol their experience separately for the phone and the watch. The results (see Figure 5.10) showed that the Wear OS users interacted more often with the Stila system than the Fitbit users. **54.5%** of Wear OS users stated that they interacted with the watch application more than twice a day. Only **15.4%** of the Fitbit user used the application more than two times a day. Interestingly, the Fitbit group interacted even less with the phone application than the Wear OS group did: **18.2%** of the Wear OS group used the phone application at least three times a day in addition to their usage of the watch application. Across both devices, Stila was used at least three times a day by **63.6%** of the Wear OS users. This corresponds to an increase of **48.2%** compared to the Fitbit group ($p = 0.0087$). The results suggest that the usage of a smart watch with the Stila system also increased the user engagement on the phone.

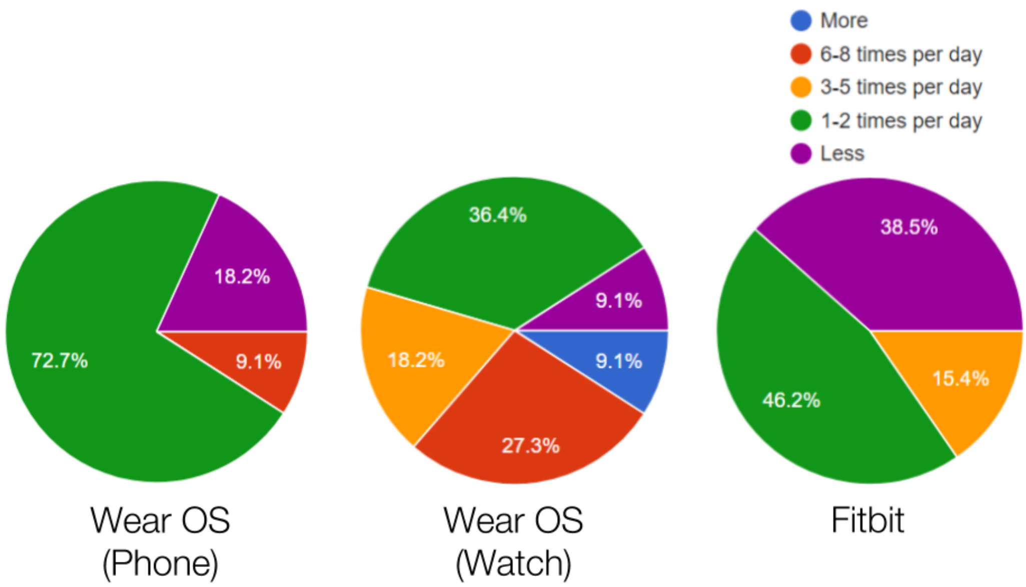

Figure 5.10: Answers to the question “How often did you interact with the App?”

The participants were also asked to specify how often they checked their stress levels on either of the devices. The results of this question confirmed that Wear OS users checked their stress levels far more often than Fitbit users: **72.7%** of Wear OS users checked their stress levels at least three times a day on the watch. **9.1%** of Wear OS participants also checked it more than twice on the phone as well. Across both devices **73.0%** of the Wear OS group checked their stress levels at least three times per day. Only **7.7%** of Fitbit users stated that they performed more than two daily stress checks. Again, results for the Wear OS users are better independent of the device ($p < 0.001$). Figure 5.11 shows the given answers as pie charts.

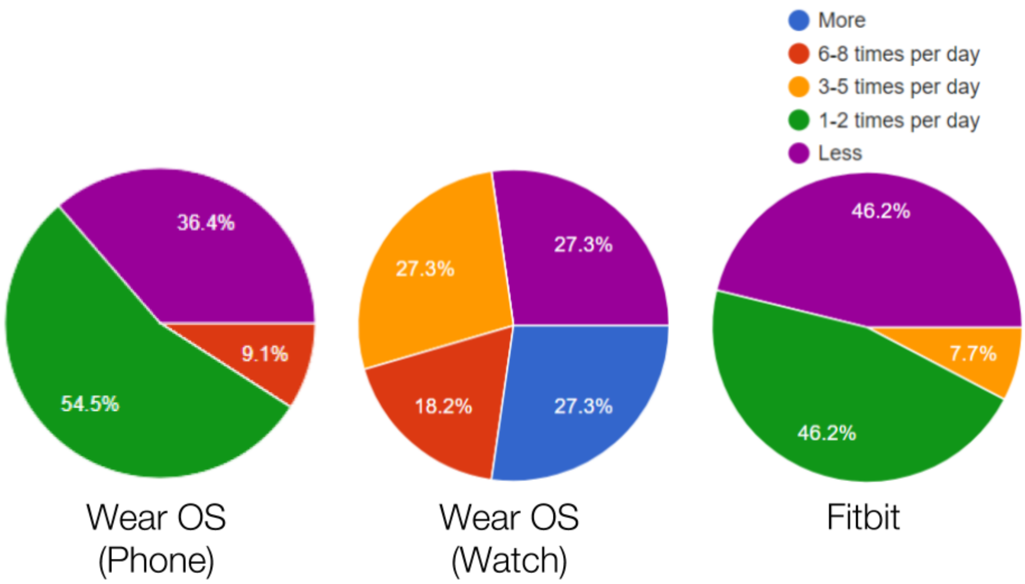

Figure 5.11: Answers to the question "How often did you check your stress levels?"

Triggers

The subjects were asked whether the notifications to track their activities came at convenient times. This question examines the functionality of the smart notifications. The Wear OS group gave the timing of the triggers an average score of **3.72** on the Likert scale. The Fitbit group rated their notifications with a significantly ($p < 0.001$) lower average score of **2.08**. This means that the Fitbit group felt that the notifications came at inconvenient moments.

Stress Awareness

Participants indicated their subjective stress awareness by rating the statement "Project Stila helped me to be more aware of my stress levels" on the aforementioned Likert scale. The Wear OS group confirmed this statement with an average score of **3.45**. Fitbit users seemed to disagree with this statement by giving an average score of only **2.46**. This shows that subjectively the stress awareness of the Wear OS group was considerably ($p = 0.0202$) higher than that of the Fitbit group.

Furthermore, the subject rated the statement "Project Stila helped me to identify stressors in my life". Both groups did not fully confirm this statement: The Fitbit group rated this statement with an average score of **2.30** while the Wear OS group evaluated the statement with a slightly higher average score of **2.72** ($p = 0.2207$).

User Feedback

The participants of the study could give their opinions about the Stila system in free text fields.

When asked what features they were missing, **Wear OS** users mentioned a notification that would alert them when they are stressed and the ability to favorite activities. Some

Wear OS users also wished for a fully capable web version, which would let the user track activities via the browser. **Fitbit** users wished for more reminders to track their activities, in-depth stress reporting and better labeling of the stress graph.

Furthermore, the subjects were asked what they disliked in the Stila applications. **Wear OS** users predominantly mentioned the high battery usage of the watch application. Furthermore, some users disliked the number of bugs they encountered during the study: For several users, it was not possible to synchronize their data to the Stila back-end. One participant was not able to use the Stila watch application because their watch did not support Wear OS 2.0.

Multiple **Fitbit** users disapproved of the way the activity tracking was implemented: "The way to add activities throughout the day is cumbersome". Furthermore, several Fitbit users did not understand the stress graph or how the stress data was derived from the heart rates.

Both groups disliked the number of dimensions that had to be filled out during the activity tracking process. Furthermore, multiple users described the naming of dimensions as "unclear" what gave them trouble when trying to give information about their activities.

The participants were also asked what they liked about the tested applications. One **Wear OS** user praised the accuracy of the stress algorithm: "I felt it was fairly accurate so I was able to rely on its suggestion that I was stressed at times I did not realise". Another user felt a positive impact on their stress awareness: "It actually made me more self aware". The **Wear OS** application was described as "easy to use" by a participant. Furthermore, the stress complication and the stress graph and the interaction with it was liked by another participant. A participant with stress-related epilepsy had a very positive experience with the **Wear OS** application: "[...] It actually helped me avoid two epileptic seizures. Just by looking at my history and thinking of the situation I was going into before I did. So simple yet so overlooked. Very well done!".

The **Fitbit** group largely liked the ability to analyze their collected stress data: One participant liked "the ability to see my stress levels throughout the day". I sum, the general concept of the application was liked, but room for improvement was seen. Another **Fitbit** participant summarized their experience in the following way: "I think it's a very cool concept and I think with a little more friendly UI it could really be a great tool for monitoring stress."

5.5 Summary of Results

The following table provides a comparison of the most important results between the Wear OS and the Fitbit group. The results of the usage analysis (chapter 5.3) are summarized in the top part. In the bottom part, the results of the user survey (chapter 5.4) are shown.

		Wear OS	Fitbit	p-value
Usage Analysis	Unique Users	36	41	-
	Tracked Activities	657	546	-
	Mean Number of Tracked Activities per User	25.27	16.55	0.1654
	Conversion Rate	41.2%	93.7%	< 0.001
	Successful Triggers	9.9%	7.9%	0.1137
	Mean Time to Record Activity	19 seconds	16 seconds	< 0.001
User Survey	Survey Respondents	11	13	-
	Understanding of Application	4.30 of 5	4.00 of 5	0.1929
	Ability to Understand the Apps immediately	54.5%	46.2%	0.3488
	Simplicity of Learning	4.36 of 5	3.77 of 5	0.0454
	Accessibility of Important Features	4.46 of 5	3.54 of 5	0.0119
	Simplicity of Usage	Watch: 4.36 of 5	Phone: 3.67 of 5	0.0182
	Design of Watch App	4.55 of 5	-	-
	Design of Watch Face	4.00 of 5	-	-
	Preferable Way of Tracking Activities	Watch: 72.7% Phone: 18.2%	-	-
	Timing of Triggers	3.72 of 5	2.08 of 5	< 0.001
	Initial Motivation	4.46 of 5	4.54 of 5	0.3670
	Decrease of Motivation	3.09 of 5	3.31 of 5	0.3341
	Possible Continuation of Usage	90.9%	84.6%	0.3267
	Users With More Than Two Daily Interactions	63.6%	15.4%	0.0087
	Users With More Than Two Daily Stress Checks	73.0%	7.7%	< 0.001
	Improvement of Stress Awareness	3.45 of 5	2.46 of 5	0.0202
	Identification of Stressors	2.72 of 5	2.30 of 5	0.2207

Table 5.2: Summarized results of the usage analysis and the user survey

CHAPTER 6

Conclusion

6.1 Verification of Hypotheses

This thesis explored the question, how user engagement can be increased to foster stress awareness with the help of smart watches. The tracking of activities and feelings by users was identified as the most important aspect to be made more engaging in the Stila system. The tracking of activities was thus designated the target behavior. Persuasive technology and a smart watch companion application were introduced into the Stila system to improve user engagement by triggering the target behavior. To verify the achievement of this goal, three hypotheses were proposed and tested with a field study. In the following section, the hypotheses are verified by examining the results of the study.

6.1.1 Amount of Tracked Activities

Hypothesis 1: *Users of the improved Stila application track their activities more often than users of the unimproved Stila application.*

The results of the usage analytics confirm this hypothesis with a low statistical significance. The Wear OS group tracked **111** activities more. Furthermore, the average Wear OS user recorded **8.72** activities more than the average Fitbit user. This shows that in the sample the users of the improved application tracked their activities more often than the users of the unimproved application. The measured result is statistically significant at the 16.54% level of confidence. Consequently, this could mean that the Wear OS group was more engaged than the Fitbit group.

This improvement to the Stila applications can enable the creation of personalized stress classifiers in the future, which would be beneficial for the accuracy of the machine learning algorithms in the back-end.

6.1.2 Increase of Ability

Hypothesis 2: *The changes from the unimproved to the improved Stila application increase the ability of its users to track their activities.*

The confirmation of **Hypothesis 1** shows that the improved Stila application incited its users to execute the target behavior. In light of Fogg's Behavior Model, this could have either been caused by a change in the motivation or the ability of the users. The approach taken in this thesis was to increase the ability of the users. To confirm that this was achieved, both the usage analysis and the user survey can provide evidence:

First of all, the roughly equal initial motivation of the participants and the similar decrease in the motivation can be confirmed by the motivation-related sections of the user survey: The statements regarding the initial motivation and the decrease of the motivation were answered very similarly. Furthermore, the user numbers for both groups descended similarly during the time of the study. This shows that the motivation dimension has not been manipulated.

Regarding the increase in the user's ability, the different approaches that were described in chapter 3 (*Conception*) can be evaluated by using data from the usage analysis and the survey:

Reduction

The simplicity of the watch application was rated **0.69** points higher on the Likert scale than the simplicity of the phone application. This is supported by the fact that **72.7%** of Wear OS users preferred the watch over the phone for recording their activities. Users of the Wear OS group generally liked the intuitive time tracking while recording an activity. Many Fitbit users complained about the lack of such a feature. This suggests that this factor played an important role in the increase of the ability. Furthermore, the participants of the Wear OS group rated the accessibility of important features **0.92** points higher on the Likert scale than the Fitbit group. Together with the good assessment of the usability and design of the watch application and the high statistical significance of these results, this speaks for a higher ability of the Wear OS group.

Self-Monitoring

Users of the Wear OS application indicated that they used Stila far more often to check their stress levels than users of the Fitbit group did: Only **7.7%** of Fitbit users stated that they checked their stress levels more than twice per day, while **72.7%** of Wear OS users indicated that they reach that number of checks on the watch alone. This shows that the ability for self-monitoring was increased. The increase was made possible by the addition of the stress complication to the Stila watch face and the technical prerequisites to allow real-time stress tracking. Another hint at the higher self-monitoring capabilities of the Wear OS application is the **52.4%** lower conversion rate: The fact that many visits to the application do not result in the tracking of an activity shows that the application is opened for other reasons as well. It can be assumed that this includes the checking of stress levels. All measurements concerning the self-monitoring tool type showed the increase in the ability with a high statistical significance.

Tunneling

The feedback of the user survey shows that the tutorial and onboarding tunnels of the applications have been successful in increasing the ability of the users: The Wear OS users (who had access to the tunnels) rated the applications averagely **0.59** points easier to learn, while also assessing their understanding averagely **0.30** points higher than the Fitbit group. **8.2%** more of the Wear OS users could understand the purpose of the application right away. Additionally, some users of the Fitbit group stated that they would have liked to obtain more information about the operation and the functioning of the application. In sum, these results show that the tunnels were very effective in the increase of the user's ability. It can be seen that the tunneling features were statistically significant in simplifying the act of learning.

Tailoring

The results show that some tailoring features were effective in the increase of ability while others were not. The measured data in this category allowed the comparison of the engagement of users who used tailoring features to users who did not. The results show that users who customized their watch face colors or the measurement interval did not track more activities than other users. However, users who created custom activities did also averagely record **10.96** activities more. In both cases, it is not clear whether there is a causal link and the results bear a low statistical significance. The creation of different watch faces was also not one of the key success factors. The results show that the majority (**75.86%**) of Wear OS users preferred the digital watch face which speaks against the necessity of the analog watch face.

Triggers

As mentioned before, *Triggers* in Fogg's Behavior Model can also be interpreted as *Suggestion Tools* in Fogg's Functional Triad. In this thesis the Wear OS group were triggered by more sophisticated suggestion tools than the Fitbit group: The smart notifications on the watch could consider the user's current stress level, while the ordinary notifications in the Fitbit version could not. The results reflect that **1.97%** more of the notifications on the watch led to the tracking of an activity. Furthermore, the users of the Wear OS group rated the timing of the notification with a **1.64** higher score than the Fitbit group. While the difference in the rate of successful triggers has a rather low statistical significance, the differences in the subjective rating of the triggers by the users is significant beyond the 99% level of confidence.

In sum, it can be seen that the improved application increased the ability of its users. Especially the reduction features and the tunnels seem to be the most successful additions. However, the measured data does not allow definite conclusions about the importance of each factor in the increase of the user's engagement.

6.1.3 Increase of Stress Awareness

Hypothesis 3: *The improved Stila application increases the subjective stress awareness of its users compared with the unimproved version.*

The results show that Wear OS users rated the subjective improvement of stress awareness averagely **0.99** points higher on the Likert scale than the Fitbit users. This means that users of the improved application felt a stronger increase in their stress awareness than users of the unimproved version. As mentioned above, **72.7%** of the watch users stated that they checked their stress levels more than twice a day, while only **7.7%** of Fitbit users did the same. It can be assumed that these stress checks are the basis for a high stress awareness.

The results of the usage analysis reinforce these statements: The Wear OS group visited the application almost **three times** as often. Together with the relatively low conversion rate of **41.24%**, this suggests that the Wear OS group used the application more often for stress checks. All survey results concerning the stress awareness have a high statistical significance.

Furthermore, it can be assumed that the aforementioned increase in tracked activities helped the user to differentiate between eustress and distress by protocoling their feelings during the activities. Subsequently, users can potentially minimize the execution of activities that they often linked to bad feelings. Hence, distress can be avoided more often.

In sum, this shows that the application encouraged the Wear OS group to increase their stress awareness.

6.2 Discussion

6.2.1 Validity

This study consisted of a software-assisted usage analysis and a user survey. The sample was composed of participants of different age groups and cultural backgrounds resulting in a large heterogeneity. Based on this sample, all hypotheses could be verified. Even though some results had a slightly higher alpha error, the results of the usage analysis could be confirmed by the user survey and vice versa. This confirmation speaks for the validity of the presented results.

6.2.2 Implications

The study results confirmed the assumption that the described persuasive improvements to the Stila system increased the user engagement by creating behavior changes. Further explorations suggested that this also has positive implications on the user's stress awareness.

Against the backdrop of this thesis' theoretical framework, the results suggest that wearables are especially well suited for the application as persuasive tools and are capable of increasing the ability of its users. Generally, wearables seem to have a higher persuasiveness than ordinary computers. This persuasiveness enables them to create a higher user engagement despite their smaller display size.

The described persuasive features have promising implications in the university environment: Increased stress awareness can help students to perform better. Better stress detection in Stila could be used to enhance lectures with live computer-derived physiological feedback from the students.

6.2.3 Limitations

The small (and unsteady) sample size limits the validity and statistical significance. The motivation of the subjects was not manipulated in this study. This led to a decrease in users. Reasons for this could include the weak long term engagement of wearables (see chapter 2.4.6 *Persuasive Wearables*) among others. The conduction of another study with more participants that are equally motivated during the study (e.g., by monetary incentives) could explore the ability dimension deeper. Furthermore, a higher number of survey respondents would have increased the statistical significance of the survey results.

Another limitation of this study is the lack of adequate units and scales to measure the user's ability and motivation. Because of this limitation, proxy measures had to be used to estimate the users' ability and motivation. These proxies lead to the fact that subtle differences or causal relationships can not always be discovered. Furthermore, the added features could not be definitively rated in their individual importance. To achieve this the features of the watch would have had to been tested individually. This study could only prove, that all the features together generated an increase in the ability of its users. It could not identify which features were most responsible. Nevertheless, the hypotheses of this study could be verified because the data of the usage analysis and the user survey showed the superiority of the watch application very clearly.

It is not clear, how much the watch decreased the access time to the application, as this could not be measured. It could, however, be seen, that users averagely take three seconds longer to track their activities on the watch as opposed to the phone. To simplify the usage of the application, the access time would have to be reduced more than three seconds.

A problem of the Stila system is the fact that users of the improved application were subjectively not better at finding stressors than users of the unimproved application. The simplified finding of (hidden) stressors is also one of the goals of Stila. These results suggest that the ability to find stressors might not be related to the amount of user engagement.

6.3 Future Work

Future research could explore the Stila system concretely or persuasive technology in general:

6.3.1 Stila

To further increase the user engagement in the Stila applications and therefore the caused stress awareness, several approaches could be taken.

Motivation

As mentioned before, this thesis only manipulated the ability dimension in Fogg's Behavior Model. To increase the engagement more, the motivation dimension could be manipulated as well. The results of this study have also shown, that the motivation of the participants seemed to decrease during the study. Therefore, it would be advantageous to add features that retain the high initial motivation of the users. An example would be the addition of gamification features. Gamification is the "use of game design elements in non-game contexts" [10]. Gamification can be used to influence human motivation and behavior in general [62].

In the context of the Stila system, it would, for example, be possible to implement gamification in the form of "badges". Users who execute the target behavior regularly get rewarded with collectible icons in the application. Badges can be effective progress markers or be used as a way for users to show off their accomplishment [62].

Another way to increase the user's engagement would be to increase her intrinsic motivation. Intrinsic motivation does not rely on an external reward [25]. A way to do this would be to offer the user more in return for using Stila. E.g., the user could be offered an explanation of mindfulness techniques when Stila registers a surge in her stress levels. This intrinsic reward of being able to cope better with her stress could motivate the user to use the application more often.

Measurement and Tracking

This thesis has concentrated on the increase in the number of tracked activities by the user. Another way to accomplish this increase would be by implementing an automated activity tracking mechanism. It is conceivable that future iterations of Stila will be able to identify some activities such as running or walking automatically. Automated activity tracking would lessen the need for user-generated activity tracking and further increase the number of recorded activities.

Future research could also explore other alternatives to heart rate variability based stress detection. Especially electroencephalography (EEG) could be used to identify stress. EEG data show the fluctuation of electrical activity in the brain. Certain electrical signals can be associated with mental processes such as focus or anxiety [56]. Wearables like the aforementioned *muse* headband could be used to measure stress with EEG.

Personalized Classifiers

As mentioned above, the higher amount of tracked activities per user could make the creation of personalized stress classifiers possible. This feature could be added to the Stila back-end to further improve the stress awareness of the platform's users.

Implementation

It would be desirable to include the results of the back-end's stress classifier into the Stila Android application. The results of the classifier would allow for very detailed stress feedback that could also include the differentiation between eustress and distress.

Furthermore, the application could be enhanced with features that allow the live-streaming of stress data to other platforms. This would allow the use of Stila in university lectures and other situations that could be improved with knowledge of the participants' stress levels.

6.3.2 Persuasive Technology

Live Interaction

Ramachandran and Canny [47] suggest that real live interaction has a higher persuasive power than technology assisted persuasion. To replicate real-life interaction, they suggest using speech-based interfaces. Especially persuasive wearables could be improved with natural language interfaces. E.g., smart watches could overcome their disadvantages (small display, difficult interaction, etc.) with the use of speech-based interaction.

Persuasive Power of Wearables

Regarding the results of this thesis, future research should examine whether wearables have a higher persuasive power under all circumstances or only in certain situations. While the findings of this study are consistent with this assumption, more studies have to be conducted to investigate this area.

List of Figures

List of Tables

Bibliography

[1] Daniel L. Ashbrook et al. "Quickdraw: The Impact of Mobility and On-body Placement on Device Access Time". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '08. New York, NY, USA: ACM, 2008, pp. 219–222.

[2] Eric Bergman. *Information Appliances and Beyond*. Ed. by Tim S. Kelly et al. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2000.

[3] Kerry Bodine and Francine Gemperle. "Effects of Functionality on Perceived Comfort of Wearables". In: *Seventh IEEE International Symposium on Wearable Computers*. White Plains, NY, USA: IEEE Press, 2003, pp. 57–60.

[4] Sansanee Boonnithi and Sukanya Phongsuphap. "Comparison of Heart Rate Variability Measures for Mental Stress Detection". In: *2011 Computing in Cardiology*. CinC 2011. 2011, pp. 85–88.

[5] Longbing Cao. "In-depth Behavior Understanding and Use: The Behavior Informatics Approach". In: *Information Sciences* 180.17 (2010), pp. 3067–3085.

[6] Xiang Chen et al. "Duet: Exploring Joint Interactions on a Smart Phone and a Smart Watch". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '14. New York, NY, USA: ACM Press, 2014, pp. 159–168.

[7] Sunny Consolvo et al. "Goal-setting Considerations for Persuasive Technologies That Encourage Physical Activity". In: *Proceedings of the 4th International Conference on Persuasive Technology*. Persuasive '09. New York, NY, USA: ACM Press, 2009, 8:1–8:8.

[8] *Creative Vision*. Wear OS by Google. URL: `https://designguidelines.withgoogle.com/wearos/` (visited on 07/07/2018).

[9] Laura Dennison et al. "Opportunities and Challenges for Smartphone Applications in Supporting Health Behavior Change: Qualitative Study". In: *Journal of Medical Internet Research* 15.4 (2013), pp. 62–73.

[10] Sebastian Deterding et al. "From Game Design Elements to Gamefulness: Defining "Gamification"". In: *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments*. MindTrek '11. New York, NY, USA: ACM Press, 2011, pp. 9–15.

[11] Fred I. Dretske. *Explaining Behavior: Reasons in a World of Causes*. Cambridge, MA, USA: MIT Press, 1988.

[12] Jesse Feiler. *Introducing SQLite for Mobile Developers*. The expert's voice in mobile programming. Berkeley, CA, USA: Apress, 2015.

[13] B. J. Fogg. "A Behavior Model for Persuasive Design". In: *Proceedings of the 4th International Conference on Persuasive Technology*. Persuasive 2009. Claremont, CA, USA: ACM Press, 2009, 40:1–40:7.

[14] B. J. Fogg. *Persuasive Technology: Using Computers to Change What We Think and Do*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2002.

[15] B. J. Fogg. *What Causes Behavior Change?* B.J. Fogg's Behavior Model. URL: http://behaviormodel.org/index.html (visited on 07/12/2018).

[16] B. J. Fogg and Jason Hreha. "Behavior Wizard: A Method for Matching Target Behaviors with Solutions". In: *Persuasive Technology*. Persuasive 2010. Berlin, Germany: Springer Berlin Heidelberg, 2010, pp. 117–131.

[17] Patrik Hagen. "Eustress and Distress Detection from Physiological Data Using Supervised Machine Learning". Master's Thesis. Munich, Germany: Ludwig-Maximilian University München, 2017.

[18] Sajanee Halko and Julie A. Kientz. "Personality and Persuasive Technology: An Exploratory Study on Health-Promoting Mobile Applications". In: *Persuasive Technology*. Persuasive 2010. Berlin, Germany: Springer, Berlin, Heidelberg, 2010, pp. 150–161.

[19] Marja Harjumaa and Harri Oinas-Kukkonen. "Persuasion Theories and IT Design". In: *Persuasive Technology*. Persuasive 2007. Berlin, Germany: Springer, Berlin, Heidelberg, 2007, pp. 311–314.

[20] Jan Hartmann, Alistair Sutcliffe, and Antonella De Angeli. "Investigating Attractiveness in Web User Interfaces". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '07. New York, NY, USA: ACM Press, 2007, pp. 387–396.

[21] Marcel Heil. "Conception and Implementation of a Mobile Application with Fitness Trackers as Supportive Tools for Computed Stress Detection". Bachelor's Thesis. Munich, Germany: Ludwig-Maximilian University München, 2016.

[22] *How does Muse work?* Muse: The brain sensing headband. URL: http://www.choosemuse.com/how-does-muse-work/ (visited on 07/18/2018).

[23] *How Glass Can Help*. Glass. URL: https://x.company/glass/ (visited on 07/14/2018).

[24] *Huawei Watch 2 - Made for Free Spirits*. Huawei. URL: https://consumer.huawei.com/us/wearables/watch2/ (visited on 08/08/2018).

[25] Atreyi Kankanhalli et al. "Gamification: A New Paradigm for Online User Engagement." In: *Proceedings of the 33rd International Conference on Information Systems*. ICIS 2012. Orlando, FL, USA: Association for Information Systems, 2012, pp. 3573–3582.

[26] Armağan Karahanoğlu and Çiğdem Erbuğ. "Perceived Qualities of Smart Wearables: Determinants of User Acceptance". In: *Proceedings of the 2011 Conference on Designing Pleasurable Products and Interfaces*. DPPI '11. New York, NY, USA: ACM Press, 2011, 26:1–26:8.

[27] Dan Ledger and Daniel McCaffrey. *Inside Wearables: How the Science of Human Behavior Change Offers the Secret to Long-Term Engagement*. Medium. 2014. URL: https://blog.endeavour.partners/inside-wearable-how-the-science-of-human-behavior-change-offers-the-secret-to-long-term-engagement-a15b3c7d4cf3 (visited on 06/07/2018).

[28] Hee Yun Lee et al. "Mobile Phone Text Messaging Intervention for Cervical Cancer Screening: Changes in Knowledge and Behavior Pre-Post Intervention". In: *Journal of Medical Internet Research* 16.8 (2014), pp. 45–55.

[29] *Levi's Commuter X Jacquard By Google - Dark Wash*. Levi's US Webshop. URL: https://www.levi.com/US/en_US/clothing/men/outerwear/levis-commuter-x-jacquard-by-google/p/286600000 (visited on 07/14/2018).

[30] Kent Lyons. "What can a Dumb Watch Teach a Smartwatch?: Informing the Design of Smartwatches". In: *Proceedings of the 2015 ACM International Symposium on Wearable Computers*. ISWC '15. New York, NY, USA: ACM Press, 2015, pp. 3–10.

[31] Regan Mandryk, Kori M. Inkpen, and Thomas W. Calvert. "Using Psychophysiological Techniques to Measure User Experience with Entertainment Technologies". In: *Behaviour & IT* 25 (2006), pp. 141–158.

[32] Steve Mann. "An Historical Account of the 'WearComp' and 'WearCam' Inventions Developed for Applications in 'Personal Imaging'". In: *Digest of Papers. First International Symposium on Wearable Computers*. ISWC 1997. Los Alamitos, CA, USA: IEEE Press, 1997, pp. 66–73.

[33] T.L. Martin. "Time and Time Again: Parallels in the Development of the Watch and the Wearable Computer". In: *Proceedings of the 6th IEEE International Symposium on Wearable Computers*. ISWC '02. Washington, DC, USA: IEEE Press, 2002, pp. 5–11.

[34] Siobhan K McMahon et al. "Older Adults' Experiences Using a Commercially Available Monitor to Self-Track Their Physical Activity". In: *JMIR mHealth and uHealth* 4.2 (2016), pp. 593–602.

[35] Cees Midden et al. "Using Persuasive Technology to Encourage Sustainable Behavior". In: *8th Biennial Conference on Environmental Psychology*. EnvPsych09. Zurich, Switzerland: Pabst Science Publishers, 2009.

[36] Gerald R. Miller. "On Being Persuaded: Some Basic Distinctions". In: *The Persuasion Handbook: Developments in Theory and Practice*. Ed. by James Price Dillard and Michael Pfau. Thousand Oaks, CA, USA: SAGE Publications, Inc., 2002, pp. 3–16.

[37] Sanjay M. Mishra. *Wearable Android: Android Wear and Google FIT App Development*. New Jersey, NY, USA: Wiley, 2015.

[38] *moto 360 - Choose the Watch that Makes Time for you*. Motorola. URL: `https://www.motorola.com.au/products/moto-360` (visited on 07/14/2018).

[39] Heather L. O'Brien and Elaine G. Toms. "What is User Engagement? A Conceptual Framework for Defining User Engagement with Technology". In: *Journal of the American Society for Information Science and Technology* 59.6 (2008), pp. 938–955.

[40] Harri Oinas-Kukkonen and Marja Harjumaa. "Persuasive Systems Design: Key Issues, Process Model, and System Features". In: *Communications of the Association for Information Systems* 24.1 (2009), pp. 485–500.

[41] Harri Oinas-Kukkonen and Marja Harjumaa. "Towards Deeper Understanding of Persuasion in Software and Information Systems". In: *Proceedings of the First International Conference on Advances in Computer-Human Interaction*. ACHI '08. Washington, DC, USA: IEEE Press, 2008, pp. 200–205.

[42] Christopher Peters, Ginevra Castellano, and Sara de Freitas. "An Exploration of User Engagement in HCI". In: *Proceedings of the International Workshop on Affective-Aware Virtual Agents and Social Robots*. AFFINE '09. New York, NY, USA: ACM Press, 2009, 9:1–9:3.

[43] Richard Petty and John Cacioppo. *Communication and Persuasion: Central and Peripheral Routes to Attitude Change*. New York, NY, USA: Springer New York, 1986.

[44] Whitney Quesenbery. "Dimensions of Usability: Defining the conversation, Driving the Process". In: *Proceedings of the UPA 2003 Conference*. UPA 2003. Houston, TX, USA: Usability Professionals Association, 2003, pp. 23–27.

[45] M. T. Raghunath and Chandra Narayanaswami. "User Interfaces for Applications on a Wrist Watch". In: *Personal and Ubiquitous Computing* 6.1 (2002), pp. 17–30.

[46] U. Rajendra Acharya et al. "Heart Rate Variability: A Review". In: *Medical & Biological Engineering & Computing* 44.12 (2006), pp. 1031–1051.

[47] Divya Ramachandran and John Canny. "The Persuasive Power of Human-Machine Dialogue". In: *Proceedings of the 3rd International Conference on Persuasive Technology*. PERSUASIVE '08. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 189–200.

[48] Ashok Kumar S. *Android Wear Projects : Create Smart Android Apps For Wearables*. Birmingham, UK: Packt Publishing, 2017.

[49] Hans Selye. "Stress without Distress". In: *Psychopathology of Human Adaptation*. Ed. by George Serban. Boston, MA, USA: Springer US, 1976, pp. 137–146.

[50] Nandita Sharma and Tom Gedeon. "Objective Measures, Sensors and Computational Techniques for Stress Recognition and Classification: A Survey". In: *Computer Methods and Programs in Biomedicine* 108.3 (2012), pp. 1287–1301.

[51] Kirk H. Shelley. "Photoplethysmography: Beyond the Calculation of Arterial Oxygen Saturation and Heart Rate". In: *Anesthesia & Analgesia* 105.6 (2007), pp. 31–36.

[52] Ben Shneiderman. "Designing for Fun: How Can We Design User Interfaces to Be More Fun?" In: *Interactions* 11.5 (2004), pp. 48–50.

[53] Ben Shneiderman. *Designing the User Interface: Strategies for Effective Human-Computer-Interaction*. Reading, MA, USA: Addison Wesley Longman, 1998.

[54] Andreas Spahn. "And Lead Us (not) into Persuasion. . . ? Persuasive Technology and the Ethics of Communication". In: *Science and Engineering Ethics* 18.4 (2012), pp. 633–650.

[55] T. Starner. "The Challenges of Wearable Computing: Part 2". In: *IEEE Micro* 21.4 (2001), pp. 54–67.

[56] Scott Sullivan. *Designing for Wearables*. Sebastopol, CA, USA: O'Reilly Media, 2017.

[57] Henri Tajfel. *Social Identity and Intergroup Relations*. Cambridge, UK: Cambridge University Press, 1982.

[58] *Wear OS by Google Smartwatches*. Wear OS. URL: `https://wearos.google.com/` (visited on 05/23/2018).

[59] E. Webster et al. "Predicting Cognitive States from Wearable Recordings of Autonomic Function". In: *IBM Journal of Research and Development* 61.2 (2017), 2:1–2:11.

[60] Mark Weiser. "Some Computer Science Issues in Ubiquitous Computing". In: *Mobile Computing and Communications Review* 3.3 (1999), pp. 12–22.

[61] Cheng Xu and Kent Lyons. "Shimmering Smartwatches: Exploring the Smartwatch Design Space". In: *Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction*. TEI '15. New York, NY, USA: ACM Press, 2015, pp. 69–76.

[62] Gabe Zichermann and Christopher Cunningham. *Gamification by Design: Implementing Game Mechanics in Web and Mobile Apps*. Sebastopol, CA, USA: O'Reilly Media, 2011.

Appendix

Questionnaire

The questionnaires were generated using Google Forms. A link to the document was sent to the participants via email. The following questionnaire was made for Wear OS users. The Fitbit group received an identical questionnaire without the sections concerning the watch application.

Stila Field Study (Wear OS)

Thank you for your participation in the Stila user study. We appreciate your interest in our research project very much.

We are interested in your experience with the app, no matter if you are still using it or stopped during the study.
Please tell us your opinion in the following short questionnaire.
The survey consists of 36 questions, 30 of which are required to answer.
It takes approximately 10 minutes to answer the questionnaire.

The following survey comprises questions and statements.
Please answer the questions truthfully and rate how much you agree with the statements.

Your data will be evaluated and possibly published in a scientific paper anonymously. The Email Address is only necessary to connect your opinions with collected usage data in the app. Please enter the address of the Google Account you used to log into Stila.

If you have any question regarding this questionnaire, please contact us: stress-studie@protonmail.com

* Required

1. **Email address** *

General Information

Please tell us about yourself

2. **How old are you?** *
   Mark only one oval.
   - 18-23
   - 24 - 29
   - 30 - 35
   - 36 - 40
   - 40+

3. **What gender do you identify with?** *
   Mark only one oval.
   - Female
   - Male
   - Other

4. **Are you a university student?** *
   Mark only one oval.
   - Yes
   - No

5. **Do you suffer from any stress-related health issues?** *
Mark only one oval.
- Yes
- No
- Maybe
- I prefer not to tell

6. **Why did you participate in this study?** *
Mark only one oval.
- General interest in stress data
- Hope to improve stress-related health issues
- Interest in technology
- Other: ____________________

7. **How did you find out about this study?** *
Mark only one oval.
- Reddit
- University Newsletter
- Facebook
- quantifiedself.com
- University Lecture
- Other: ____________________

Stress
Please answer these questions regarding the impact of stress in your life

8. **My life is stressful in general** *
Mark only one oval.

	1	2	3	4	5	
strongly disagree	◯	◯	◯	◯	◯	strongly agree

9. **My life was stressful during the time of the study** *
Mark only one oval.

	1	2	3	4	5	
strongly disagree	◯	◯	◯	◯	◯	strongly agree

10. **What elements in your life cause you stress?** *
Check all that apply.
- [ ] University
- [ ] Job
- [ ] Personal Life
- [ ] Other: ____________________

User Experience

Please rate the following statements regarding your user experience.

11. **I fully understood the purpose of the Stila App** *
Mark only one oval.

	1	2	3	4	5	
strongly disagree	◯	◯	◯	◯	◯	strongly agree

12. **When did you understand the purpose of the Stila App?**
Mark only one oval.

◯ Right Away
◯ After the first few usages
◯ After several days
◯ Never
◯ I do not know

13. **Stila was easy to learn** *
Mark only one oval.

	1	2	3	4	5	
strongly disagree	◯	◯	◯	◯	◯	strongly agree

14. **The Stila WATCH app is easy to use** *
Mark only one oval.

	1	2	3	4	5	
strongly disagree	◯	◯	◯	◯	◯	strongly agree

15. **The Stila PHONE app is easy to use** *
Mark only one oval.

	1	2	3	4	5	
strongly disagree	◯	◯	◯	◯	◯	strongly agree

16. **The important features are quickly accessible** *
Mark only one oval.

	1	2	3	4	5	
strongly disagree	◯	◯	◯	◯	◯	strongly agree

17. **The app notifies me at convenient times** *
Mark only one oval.

	1	2	3	4	5	
strongly disagree	◯	◯	◯	◯	◯	strongly agree

18. **It is easy to navigate the watch app** *
Mark only one oval.

	1	2	3	4	5	
strongly disagree	◯	◯	◯	◯	◯	strongly agree

19. **My motivation to use the Stila app at the beginning of the study was high** *
Mark only one oval.

	1	2	3	4	5	
strongly disagree	◯	◯	◯	◯	◯	strongly agree

20. **My motivation to use the app decreased during the study** *
Mark only one oval.

	1	2	3	4	5	
strongly disagree	◯	◯	◯	◯	◯	strongly agree

Design

Please rate the following statements regarding the design of the watch app.

21. **I like the design of the watch app** *
Mark only one oval.

	1	2	3	4	5	
strongly disagree	◯	◯	◯	◯	◯	strongly agree

22. **I like the design of the watch faces** *
Mark only one oval.

	1	2	3	4	5	
strongly disagree	◯	◯	◯	◯	◯	strongly agree

23. **The buttons and texts on the watch were big enough** *
Mark only one oval.

	1	2	3	4	5	
strongly disagree	◯	◯	◯	◯	◯	strongly agree

Usage

Please answer the following questions regarding your usage of the app. Note that some question concern the Stila Phone App and others the Stila Watch app.

24. **How often did you interact with the PHONE app?** *
Mark only one oval.

- More
- 5-7 times per day
- 3-5 times per day
- 1-2 times per day
- Less

25. **How often did you interact with the WATCH app?** *
Mark only one oval.

- More
- 6-8 times per day
- 3-5 times per day
- 1-2 times per day
- Less

26. **How often did you check your stress levels on your PHONE?** *
Mark only one oval.

- More
- 6-8 times per day
- 3-5 times per day
- 1-2 times per day
- Less

27. **How often did you check your stress levels on your WATCH?** *
Mark only one oval.

- More
- 5-7 times per day
- 3-5 times per day
- 1-2 times per day
- Less

28. **Which features did you find useful on the watch?** *
Check all that apply.

- [ ] Stress Graph
- [ ] Stress Complication
- [ ] Activity Tracking
- [ ] Push Notification
- [ ] None
- [ ] Other: ______________________

29. **Was it more convienient to record your activities on the watch or on the phone?** *
Mark only one oval.

- Watch
- Phone
- Same
- I do not know

30. **Why did you find it easier to record your activities on the watch/phone?**

Stress Awareness

Please rate the following statements regarding your stress awareness

31. **Project Stila helped me to be more aware of my stress levels** *
Mark only one oval.

	1	2	3	4	5	
strongly disagree						strongly agree

32. **Project Stila helped me to identify stressors in my life** *
Mark only one oval.

	1	2	3	4	5	
strongly disagree						strongly agree

Feedback

Please support Project Stila by giving us additional Feedback

33. **Were there any features you missed in the Watch or Phone App?**

34. **What did you like in Stila?**

35. **What did you dislike in Stila?**

36. **If you stopped using the app before the end of the study, please tell us why**
Check all that apply.

- [ ] Too difficult to use
- [ ] Not enough motivation
- [ ] Not enough benefits
- [ ] Forgot to use it
- [ ] Technical issues in Stila
- [ ] Other:

37. **Will you continue your usage of the Stila Apps?** *
Mark only one oval.

- Yes
- No
- Maybe
- If I acquire a compatible heart rate monitor in the future

38. **Any comments you would like to share?**