Real time posture tracking using Breathline

Posture, body position and movement classification using an Arduino, accelerometer and k-Nearest Neighbour classification.

Graduation Project

Author:
Martijn Poot - s1858270

Supervisor:
Faber, Erik, dr.ir.
Critical observer:
Salm, Cora, dr.ir.

Study:
Creative Technology, Bsc

Faculty:
Electrical Engineering, Mathematics and Computer Science (EEMCS)

UNIVERSITY OF TWENTE.

Enschede

July 3, 2020

ABSTRACT

Breathing is perhaps the most underappreciated function of the human body and mind, it influences not only if we are alive or dead but also our state of mind. Breathing techniques have been practiced by individuals in practices such as Tai-Chi, Qigong and Yoga throughout history. Abdominal breathing is seen as the most natural way of breathing and in order to train this the Breathline wearable was developed.
The Breathline wearable makes use of Respiratory Inductance Plethysmography (RIP) to measure breathing and give the user feedback on their performance. Because it is unknown what the influence of posture, body position and movement is on abdominal breathing the clients of this project wanted to know if it is possible to integrate posture classification into the Breathline wearable. In order to solve this the following research question was formulated: ”How can a wearable mounted in the abdominal region be used to detect and classify body position, posture and movement data?”.

Background research showed that breathing is influenced by body position and verified the need to integrate posture classification in the Breathline wearable, further research on available sensors concluded that IMU’s are the most viable solution with accelerometers as a close second. A system mounted in the abdominal region making use of an Arduino and an accelerometer is then envisioned and various methods for classification are explored.
A Hi-Fi prototype is built making use of k-Nearest Neighbour classification and user tests are performed to validate the accuracy of the prototype as well as to explore the influence of posture and body position on breathing.

The prototype system created performs with an accuracy of >90% when classifying posture, body position and movement. No correlations between different postures and respiratory rate and amplitude were found but this could be because of the small sample size, further more in depth research is recommended to find conclusive results.

ACKNOWLEDGEMENTS

I would like to thank several people for their motivation, time, support and laughter towards and throughout this project.
I would like to start things of with my supervisor Erik Faber, for the countless meetings where he managed to motivate me despite the struggles I ran into, and the valuable feedback he provided on the work I produced.
Second I would like to thank the clients Ben Bulsink and Parviz Sassanian for providing me with the opportunity to work with the Breathline wearable and for meeting with me multiple times to discuss my solutions and results.
Furthermore I would like to thank my critical observer Cora Salm for taking the time to write feedback and support for this project.
Of course all my thanks go out to my roommates who were the participants of my user testing during these times of corona, including the many jokes we had about me observing them during their ”daily” tasks.

Finally, my love goes out to my girlfriend, without you I would probably not have finished this project in time.

Contents

1 INTRODUCTION

This first section aims to introduce the Breathline project and the problem which will be addressed in this graduation project. Following this will be a description of the important terminology and concepts, the research question and sub research questions and finally a description of the structure of the project.

1.1 Problem description

Abdominal breathing (see section 1.2.1) is stated as being the most natural and healthy way of breathing. In eastern medicine correct breathing techniques are practiced to treat a plethora of different diseases[1]. Correct breathing techniques are associated with several health benefits, it is shown to increase vital capacity in patients suffering from asthma and chronic obstructive pulmonary disease, as well as being able to reduce pains associated with migraines. Diaphragmatic breathing is also seen as one of the better techniques to achieve reductions in stress and anxiety and is shown to have a relaxing and stabilizing effect on the autonomic nervous system[2]. Costal breathing, also known as shallow breathing or upper chest breathing is often associated with health disorders such as dysfunctional breathing[3] and hyperventilation syndrome[4].

Research has also been done on the feedback patients get while doing abdominal breathing exercises and it was found that patients who get biofeedback during their breathing exercises achieve much better results[5]. The clients of this project, Ben Bulsink, an independent product developer and Parviz Sassanian, an acupuncturist and Tai Chi instructor, developed a prototype breathing sensor to measure and be able to give feedback to users called Breathline.
The Breathline wearable currently is used to measure a subject's breathing pattern using a measurement process called Respiratory Inductance Plethysmography(RIP, see section 1.2.2). This collected breathing pattern can then be compared to their ideal breathing pattern which is measured together with a professional. From the data that the wearable collects the quality of a person's breathing can be derived, where the focus lies on correct abdominal breathing.

The clients suspect that breathing pattern and the effectiveness of abdominal breathing is influenced by posture, position(sitting, standing, and laying down), and movement. And literature does confirm that while in the supine position abdominal breathing is more natural and prevalent and that while sitting chest wall movement is more prevalent[6, 7]. It is however unknown what the influence of posture, sitting up right or slouching, or movement is on abdominal breathing. In order to better classify the data that is collected via RIP the Breathline sensor was also equipped with an accelerometer. The clients are interested to see if it is possible to use the integrated accelerometer or if necessary additional sensors to classify posture, body position and movement.

1.2 Context analysis

In this section a short treatise is given on some of aspects of the Breathline wearable.

1.2.1 Abdominal breathing

Abdominal breathing (see figure 1.1), also known as diaphragmatic breathing or belly breathing is a breathing technique done by contracting the diaphragm muscle which allows air to enter the lungs and expand the abdomen[8]. This technique increases diaphragm length and breathing efficiency and facilitates more efficient exhalation[2].

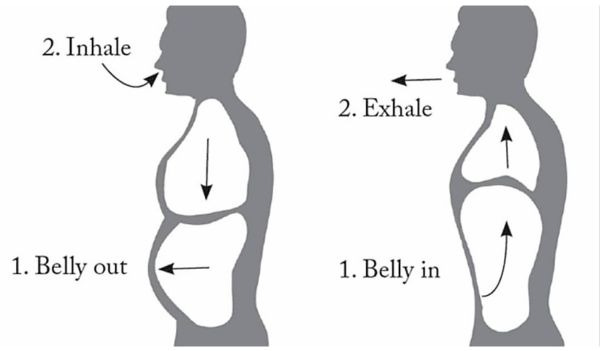

Figure 1.1: Abdominal breathing[9]

1.2.2 Respiratory Inductance Plethysmography

Respiratory Inductance Plethysmography (RIP, see figure 1.2) is a non invasive method of collecting breathing data, RIP measures the change in cross-sectional area via self-inductance of an insulated wire on an elastic band which is wrapped around the abdomen and the upper thoracic. When the band is stretched the inductance of the wire changes which can be read out and can be used to calculate the tidal volume. If both sensors are used a calibration scheme can be used to form a linear combination of the abdominal sensor and the upper thoracic sensor[10]. Cohen et al.[10] used a relatively large system which required a direct connection to a computer, Rahman et al.[11] were, because of the huge advancements in technology, able to create a fully wearable system that uses a wireless connection to send the data to an iPad. The Breathline wearable created by Ben Bulsink currently uses either the abdominal RIP sensor or both the abdominal sensor and the upper thoracic sensor. Data is stored on the wearable and can be read out by connecting the wearable to a computer via a micro USB connection.

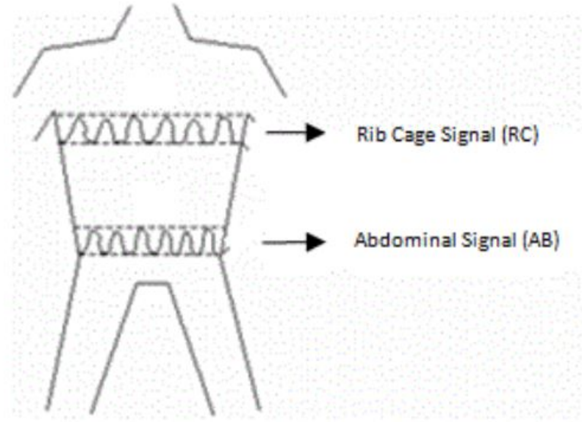

Figure 1.2: Accelerometer axis directions[12]

1.2.3 Accelerometer

Acceleration sensors or accelerometers measure movement directions (see figure 1.3), and when not moved are able to measure the inclination of the sensor with respect to gravity. They have low power consumption and are cheap to manufacture which makes that they are frequently used in wearable devices. A thorough explanation of the how accelerometers work can be found in section 2.2.1. Breathline has a single accelerometer which is mounted in the abdominal RIP sensor.

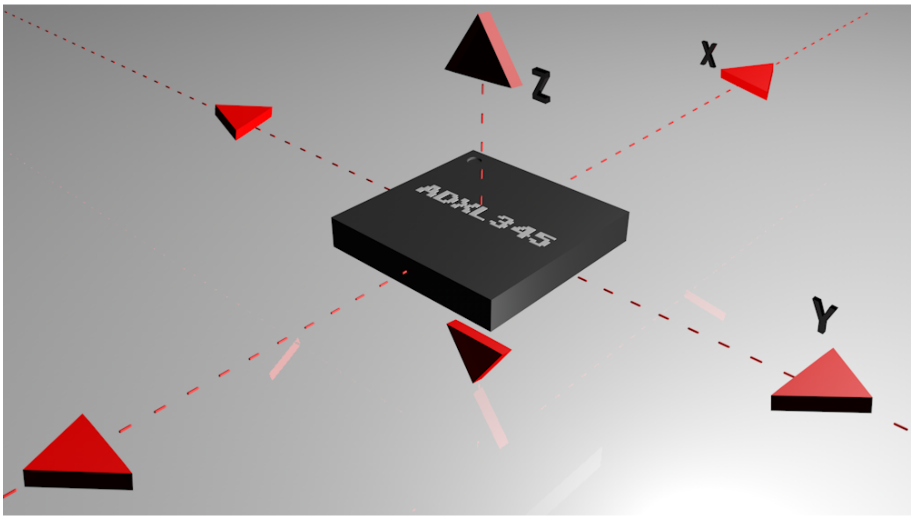

Figure 1.3: Accelerometer axis directions[13]

1.3 Goals and Research questions

1.3.1 Goals

After meeting with the clients of this project the following goals and accompanying research question were formulated.

Using the integrated accelerometer of the Breathline wearable to accurately detect and classify Body position, Posture and movement.

Integrate this data with the breathing data of Breathline. To see if there are any correlations between the two data sets.

1.3.2 Research questions

How can a wearable mounted in the abdominal region be used to detect and classify body position, posture and movement data?

To best answer this question several sub research questions are also formulated and listed here. The first pair of sub research questions are answered through a literature research and the second pair of research questions will be answered during prototyping and user testing.

What influence do posture, movement and position have on abdominal breathing?

What wearable sensors can be used to identify posture, movement and position?

What classification methods are available and which methods are usable on a small microprocessor for classifying a person's posture, movement or position?

What benefits can be gained from comparing current breathing pattern to the ideal breathing pattern when classifying body position or posture?

1.4 Report structure

Following the introduction, chapter two will go more in depth and a literature review will be conducted to answer the following two sub research questions: "What influence does posture, movement and position have on the effectiveness of abdominal breathing?" and "What wearable sensors can be used to identify a person's sitting position?". In chapter two a market research is also conducted to find existing solutions and competitors of the Breathline wearable.
In chapter three the design methodology is presented that will be used in this study followed up by the ideation phase in chapter four, where the conclusions of the background research are used to generate requirements and solutions for measuring posture, body position and movement.
This is followed by chapter five where the requirements of the ideation phase are further evaluated and specified so that it is possible to to develop a prototype in the next phase of realisation in chapter six, chapter seven covers the evaluation of said prototype and the results gathered which will be used to answer the final sub research questions: "How can an accelerometer mounted in the umbilical region be used to identify a person's movement, posture or position?"

and ”What benefits can be gained from comparing current breathing pattern to the ideal breathing pattern when identifying sitting position?”.
In chapter eight the research questions will be answered, additional findings will be discussed as well as any problems and limitations that were encountered during the project, the whole project is concluded and recommendations for further research are made.

2 BACKGROUND RESEARCH

This section provides a literature research on the influence of position, posture and movement on abdominal breathing, and viable wearable sensor options to detect and define these positions, postures and movements. Finally a preliminary conclusion is drawn on each of the subjects.

2.1 The influence of position and movement on abdominal breathing

2.1.1 Definition of positions

The positions covered in this section are the supine position, the sitting position and the standing position. The supine position means lying horizontally with torso facing up (see figure 2.1a), this is the exact opposite of the prone position, where the patient is lying horizontally with the torso facing down[14]. The sitting position means that the patient is sitting down on a horizontal surface with their back and head against a vertical surface and their feet placed on the ground (see figure 2.1b). The standing position means that the patient is standing up right with their back and head against a vertical surface (see figure 2.1c).

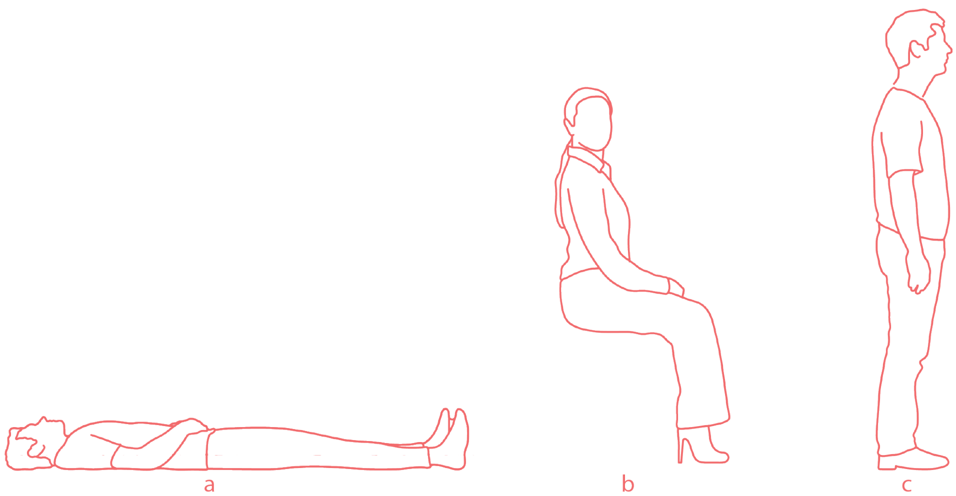

Figure 2.1: a) Supine position b) Sitting position c) Standing position [15, 16, 17]

2.1.2 The influence of position on abdominal breathing

The influence of position on breathing is widely covered in literature and their are two common ways of measuring the impact of position. The first method is through changes in breathing volume during inhaling and exhaling[18], and the second method is through external movement of the chest and abdominal region when inhaling and exhaling[7, 19], some studies cover both methods[20, 6]. Advantages of the external method are that the differences between abdominal and chest movement can be easily identified. When working with breathing volume, only the differences between total volume can be compared between the different positions, which limits the knowledge on the influence of that particular combination of position and breathing type.

The first method of measuring breathing volume during inhaling and exhaling is the preferred method when interested in volume changes. In the study of Katz et al.[18] forty-three studies are compared, those included studies with healthy subjects (29 studies) and studies with patients suffering from the following diseases or injuries: Lung disease (9 studies), spinal cord injury (7 studies), heart disease (4 studies), obesity (4 studies), and neuromuscular disease (3 studies). In this study the main focus is on the changes in forced vital capacity (FVC), forced expiratory volume in 1s (FEV1) and Total lung capacity (TLC) in different body positions of the participant. Seventeen studies were analyzed about the differences in FVC between body positions, of which four studies included subjects with a lung disease. It was found that the FVC for healthy patients was higher in a sitting position then in a supine position. In two of the studies it was also found that the FVC in standing position is also higher then in supine position. In the four studies concerning subjects with lung diseases an increase in FVC was also found, two studies found significant differences between standing and supine while only one of those studies found a significant difference between sitting and supine.
An increase in FEV1 for the sitting position when compared with the supine position was reported in seven studies and in one study an increase of FEV1 was found when comparing standing with sitting. However, in four studies there was no significant difference found between sitting and supine.
Two studies conducted on the total lung capacity found no statistically significant difference between sitting and supine position, one of the studies was conducted with both healthy subjects and subjects suffering from diabetes and the other study had only participants that suffered from diabetes.

The second method of measuring breathing through the external movements of the chest and abdominal region is the preferred method when studying the differences between chest and abdominal breathing. In the studies of Kaneko and Horie[7] and Mendes et al.[19] the effects of posture, sex and age on chest wall and abdominal movement were measured using a system of 3 dimensional measurements which use markers and cameras to track movement. In the study of Kaneko and Horie[7] participants were asked to perform both normal quiet breathing and deep breathing exercises during which their breathing movements were recorded. The effect of posture is reported to be significant during both normal and deep breathing, the chest movement was much greater in the sitting position then in the supine position under both breathing conditions. The abdominal movements were significantly less during quiet breathing than during deep breathing and were more prevalent in the supine than in the the sitting position.
When comparing the male group to the female group, the male group shows significantly less chest movement and more abdominal movement than the female group in both positions during quiet breathing, and the male group showed significantly more movement in all areas during deep breathing in both positions. Age influences the amount of chest and abdominal movement as well, it was found that older participants had less chest movement and more abdominal movement, this is probably because of less compliance of the chest wall in older participants.
It is interesting to note that respiratory rates were significantly higher in the sitting position than in the supine position.
The results of Kaneko and Horie[7] are supported by the study of Mendes et al.[19]. Mendes et al. found that posture influences all variables of the breathing pattern, and that there is a significant difference between sexes. The contribution of chest wall movement in breathing was significantly less in the supine position compared to the sitting position, and the abdominal

movement significantly increased in the supine position compared to the sitting position. It was also found that sex significantly influenced the results with the same findings as Kaneko and Horie, and that age limited the amount of chest wall movement in older participants which was then compensated with more abdominal movement.

Two of the found studies use both methods either to compare results[20] or to get the best possible data in all situations[6]. Hagman et al.[20] studied the respiratory movement measuring instrument (RMMI) which is a laser based technique to measure respiratory movements as well as breathing patterns, the objective of the study was to find whether the RMMI can discriminate between abdominal, normal and high costal (chest) breathing patterns, in three different body positions (supine, sitting and standing). Twenty participants were measured with the RMMI and a subgroup of twelve participants were also measured with a Cardio Perfect dynamic spirometer to be able to measure their breathing volumes in the diffrent body positions. The conclusion of the study is that the RMMI could differentiate between the different breathing patterns, and that the measurements showed strong correlation with the breathing volumes in all body positions.
In their study using the RMMI Hagman et al.[20] found that the contribution of the abdomen while executing an abdominal breathing pattern was 77% in the supine position, 68% in the sitting position, and 73% in the standing position, the contribution of the abdomen in natural breathing was 70% for the supine position, 53% in the sitting position and 50% for the standing position. Confirming and expanding on the results of Kaneko and Horie[7] and Mendes et al.[19] that chest movement is more prevalent in the sitting and standing positions, even while focusing on executing an abdominal breathing pattern.
The results of the literature review of Sonpeayung et al.[6] are the following: "The results of this review indicated that the sitting position improved the rib-cage compartment of the chest wall, whereas the supine position resulted in the superior enhancement in the part of the abdomen relative to other body positions". Confirming the results of other studies discussed in this literature review so far, it was also found that the volume of the abdomen is significantly lower when comparing prone and supine and also when comparing fowler and supine.

2.1.3 Definition of movement

For the scope of this literature review movement is defined as walking or running in a straight line or on a thread-mill. This definition is chosen to allow for testing in a controlled environment during the development of the Breathline sensor. There is very little research on the influence of movement on the ratio of chest and abdominal breathing but one paper was found that tests the influence of running[21] and one paper that tests the influence of inclination on the ratio of breathing[22].

2.1.4 The influence of movement on abdominal breathing

Heyde et al.[21] tested the respiratory inductance plethysmography (RIP) for its accuracy in estimating tidal volume of the lungs during standing still, running and recovery phases, a flowmeter (FM) was used to confirm ventilatory timing and volume. Only participants that had no indication of cardiovascular or lung diseases were selected, in total 98 male participants and 88 female participants took part in the study. The subjects all were familiar with the testing protocol, and the test consisted of 5 min of standing still followed by running at an increasing speed until exhaustion and than 10 min of recovery where the participant stands still.
Heyde et al.[21] found no significant evidence for a change in the ratio of chest and abdominal movement when comparing standing still, running and recovering, which means that when breathing naturally the amount of chest movement and abdominal movement increase with the

same factor. However, Bernardi et al.[22] did find a difference in chest and abdominal breathing ratio when the incline of the running surface is changed. in their study they tested 15 participants with a RIP in an uphill run and found that steep slopes led to a reduction in thoraco-abdominal coordination, specifically a reduction in rib cage displacement because of a more forward body position[22]. Again, contributing the change in breathing ratio to body position and not exercise, which matches with the conclusion of Heyde et al[21] ”As no systematic differences between calculated gains of non-exercise and exercise data were observed, further derived tidal volumes seem to be not affected by body movement during running. It has already been shown that calculated gains are posture dependent and can differ up to ±40% when changing from supine to upright posture or vice versa”.

2.1.5 Preliminary conclusion

Considering the findings of Hagman et al., Sonpeayung et al., Kaneko and Horie, and Mendes et al. it is safe to assume that there is a significant difference in breathing ratio (the ratio between abdominal and chest breathing) between the supine and the sitting position when a person is breathing naturally. A significant difference in the breathing ratio between supine and standing is also found but a significant difference in breathing ratio between standing and sitting could not be found. And from the results of Hagman et al. it is found that there also is a significant difference in breathing ratio when a person is performing abdominal breathing in either the supine or sitting position. From which it is possible to conclude that the supine position seems to be the most effective for abdominal breathing as it allows for the greatest movement in the abdominal area.
It is important to note that there was no significant volume difference between the different body positions, which means that a change in position might only account for a change in breathing ratio, however this change in position then does influence the FVC of the person. It was found that the influence of movement is not significant on the breathing ratio of a person, and that chest and abdominal movement might increase due to increased oxygen needs but there is no significant difference in ratio.

2.2 Wearable sensors

Body position and movement recognition is a hot topic in both medical and human computer interaction (HCI) research, many types of systems such as acoustic and electromagnetic recognition and computer vision have been suggested and used through the years but only very few technologies are viable for integration into a wearable system[23]. The most viable technologies for integration into a wearable system will be discussed in this section.

2.2.1 Accelerometer

Through the development of Micro Electro Mechanical Systems (MEMS) inertial sensing units were able to shrink in size and power consumption greatly. MEMS acceleration sensors (see figure 1.3) function by having a silicon mass supported by silicon springs and the displacement of the mass due to acceleration or gravity is measured by a capacitance change between the mass and the fixed electrodes that surround it[24]. There are three main types of acceleration sensors, variable capacitive MEMS, piezoresistive and piezoelectric[25]. Piezoresistive acceleration sensors are used for high impact low sensitivity applications and for the scope of this report not relevant. piezoelectric acceleration sensors are not able to measure static accelerations and sensor orientation which mean they are not relevant for the applications in this report. Variable capacitive MEMS acceleration sensors can be used to dectect motion in all three axis and output the static sensor orientation with respect to gravity[26, 27]. Accelerometers are frequently used to measure movement data[28, 29, 30, 31, 32, 33] and calculate body positions[29, 31, 32, 33, 34, 35, 36], a short overview is given of the results gathered by other researchers.

A single accelerometer mounted at the thigh of a test subject showed good results in accurately identifying between sitting and standing position, while an accelerometer mounted at the waist showed satisfactory results. However, when identifying supine position an accelerometer mounted at the waist showed good results and an accelerometer mounted at the thigh showed satisfactory results[33]. These results are confirmed by Jason et al. where an accelerometer at the waist showed good results in identifying between supine and sitting or standing, but an extra accelerometer at the thigh was needed for a clear differentiation between sitting and standing[34]. Chen and Chen also found that the waist is the best mounting position, comparing to the upper back and thigh, for an accelerometer when checking if a patient is upright or has fallen[30].
Accelerometers are also used to identify whether patients are standing still, walking or jogging. Walking and jogging were detected by a single thigh mounted accelerometer with high accuracy and with moderate and high accuracy for a single waist mounted accelerometer[33]. Gyllensten and Bonomi elaborate on this by comparing a single accelerometer system mounted at the waist with a 5 accelerometer system mounted at the feet, thighs and chest. They found that when using machine learning algorithms a single accelerometer delivers excellent results for identifying both body positions and types of movement in a lab setting. However, they found less accurate results when using those same algorithms on data gathered in a real life settings, and they explain that this is probably because daily activities differ from those in a lab setting, people sit differently, they move around more and do different activities that were not tested in the lab[32].

Three-axis accelerometers are perfect for integration into wearable devices because of their low power requirements, low cost and small size. However because of the decrease in cost and size requirements of gyroscopes it has become common to pair these two sensors into one packet called an inertial measurement unit (IMU).

2.2.2 Gyroscope

MEMS single axis gyroscope sensors are built in the same priciple as MEMS acceleration sensors, a silicon mass is supported by springs. The big difference is that angular velocity is measured by the Coriolis force on the suspended mass, however in order to measure the Coriolis force the mass has to be vibrating with at least two degrees of freedom[24, 37]. Multi axis gyroscopes are constructed in several ways which offer benefits and disadvantages and should be carefully select for the application in mind. Single axis gyroscopes can be assembled into 3 axis gyroscope by mounting a single axis gyroscope on each plane (see figure 2.2a) , these gyroscopes offer very good performance but are often large in size. Another approach is the single chip method, where either multiple single axis gyroscopes are mounted in a single plane (see figure 2.2b) or a single mechanical element measures rotation in multiple axes (see figure 2.2c). These methods both offer significantly reduced size, however can be less accurate because of a lack of symmetry in the XZ an YZ planes in the case of multiple gyroscopes or cross talk between axes in case of the single element approach[38].

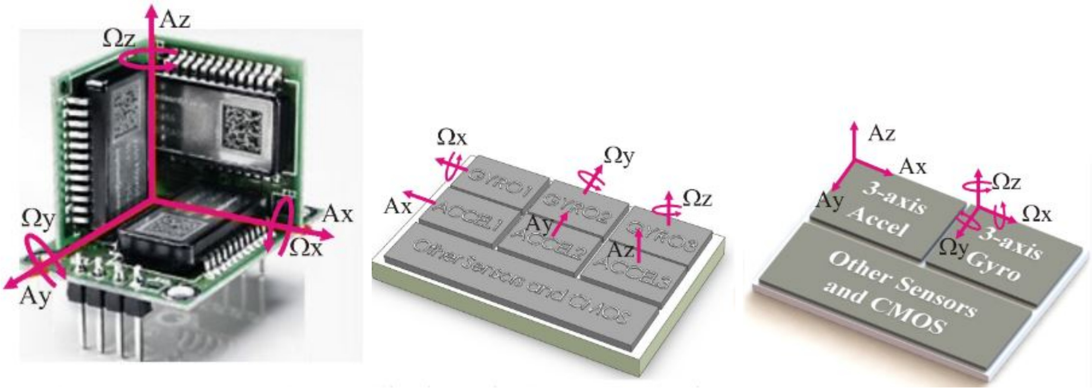

Figure 2.2: a) sensor on each plane b) multiple single axis sensors c) single element sensor [38]

A gyroscope is used to measure angular velocity, which means that changes in orientation can be measured even when the gyroscope is moving, as opposed to an accelerometer which can only calculate orientation when not moved. Gyroscopes suffer from what is called zero-point or zero bias stability, this is because gyroscopes are not able to use gravity as a reference point like accelerometers are, an accumulation of calculation errors can lead to the gyroscope being of to up to hundreds of degrees. Techniques such as Kalman filtering are being applied to correct this drift, but these techniques are not perfect[24, 39]. Which is why gyroscopes are not used much on their own, however testing on using standalone gyroscopes has been done by Mansour et al.[40].

A zero-lag Butterworth low and high pass filter was used to cancel possible offset and remove spurious peaks, and it was found that gyroscopes can be used reliably to measure heel down and toe off (gait) timing in walking situations with the gyroscope mounted on the lower leg between the ankle and knee. And they conclude that a gyroscope offers good accuracy to detect gait events[40]. From this it is possible to say that gyroscopes can offer an opportunity in detecting movement, however more effective methods are needed to counteract the zero bias stability. One such method is dicussed by Lai., they propose a combination of quaternion and a Kalman filter to detect posture using a three-axis gyroscope. The output of a three-axis accelerometer and a three-axis magnometer is used to correct the accuracy of quaternion based on a Kalman filter. And their experiments show that this method is effective and improves measurement accuracy[23]. From this research it is safe to state that a gyroscope on its own does not offer a viable solution for integration into a wearable product, however when paired with an accelerometer it can offer valuable extra data that can help detect posture when moving.

2.2.3 Inertial measurement unit

Inertial measurement units (IMUs, see figure 2.2) are self contained systems created especially for measuring movements of objects or humans. These systems contain at least an accelerometer and gyroscope[37] but a variety of other sensors and electronics can also be included, such as but not limited to magnetometers and micro-controllers[23, 41]. And since the development of MEMS sensors these systems were able to shrink in size, power consumption and price, and these factors have contributed to the increasing popularity and market size of these devices[42]. Most research done in the past years involving the detection of posture[43, 44, 45], body position [46, 47, 48, 49] and movement[46, 47, 48, 49, 50, 51] have been done using one or multiple IMUs at least containing an accelerometer and a gyroscope. And a literature review on role of wearables in spinal posure analysis revealed that IMUs are the most used wearable tool for measuring spinal posture, Simpson and Maharaj[52] reviewed 37 articles and found that the most accurate systems in determining spinal posture used IMUs. However, they also state that there is limited data regarding validation of these devices.

All research found on posture used multiple IMUs situated along the whole[43, 44] or the lower[45] spine, to measure the real time angles of the spine. And all researchers found IMUs to deliver good results in accurately measuring spinal posture. Research on body positions reveals that IMUs can detect body postion and the type of position with high accuracy[46, 49, 47, 48] and that viable single sensor locations for detecting and differentiating body positions are the pelvis[48], sternum[49, 48] and head[48]. Tanaka et al.[46] and Airaksinen et al.[47] used multiple sensor locations in their calculations. Movement detection is often considered as a simple problem to solve, but can quickly evolve into a complex problem when multiple sensors and complex algorithms are used[50]. Three papers were analyzed on detecting and differentiating movement using a single IMU[49, 50, 51], and two papers were analyzed that used multiple IMUs[46, 47], one paper compared using multiple IMUs and using one IMU[48]. Zheng et al/[50] and Yang and Li[51] analysed walking by using a single IMU mounted on the lower leg[51] and the foot[50] and they were able to identify walking speed and walking posture without problems. Najifi et al.[49] used a single sternum mounted IMU and were also able to identity whether a patient was walking, however they found that it was hard to identify fast movements. However, Vial et al.[48] found that a single pelvis mounted IMU performed just as well as a combination of IMUs mounted on the sternum and head in idenfitying different movements and they found less trouble in identifying fast movements.

2.2.4 Preliminary conclusion

IMUs are seen as the gold standard in body position and movement tracking, and can also be employed to track sitting posture, current generation IMUs are low volume and have limited power requirements, this makes them ideal for integration into a wearable product. Combined with good availability and a low price the ideal solution would be to integrate an IMU into the Breathline wearable. Accelerometers alone are a good second option, as they allow for sitting posture tracking as well as movement detection, gyroscopes are not a single sensor solution because of their zero-bias and complicated algorithms that are needed to address the problem. Important to note is that a single IMU or accelerometer mounted in the umbilical region might have difficulties tracking posture because of abdominal movement and less degrees of movement when the posture changes, compared with multiple sensors on different parts of the body, and difficulties may arise with the accelerometer when tracking types and speeds of movement.

2.3 Market research

After doing a literature search on types of wearable sensors and their application as body position, posture and movement tracking a market research is necessary to see what configurations of sensors the competing products use and at what places of the body the sensors are situated. The results contain both breathing related as well as posture related wearables and some that do both.

2.3.1 Wearable products

Prana

Prana (see figure 2.3) is a wearable which tracks both breathing and posture in order to promote better breathing and reduce stress. Prana tracks diaphragmatic breathing and posture when sitting and has two modes, a passive mode where it notifies you when either breathing or posture can be improved and an active training mode where breathing can be practised through a game. When walking Prana functions as a step tracker[53]. The Prana wearable is mounted on the inside of the waistband of the user's clothes and has a buzzer which is used to notify you of alerts as well as an application connection which can send notifications and can be used to track the user's performance. The Prana wearable contains both an accelerometer as well as a gyroscope to measure breathing and posture data, all calculations of the data are done on an external device connected via Bluetooth[54] for example a smartphone.

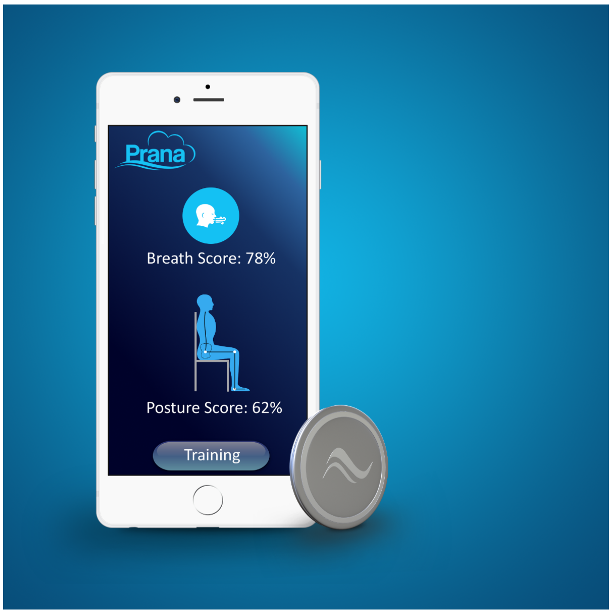

Figure 2.3: Prana[53]

BreathBalanz

BreathBalanz (see figure 2.4) is a respiratory measurement wearable aimed at achieving healthy abdominal breathing, the application that comes with the wearable features an 80-day train program focused on working towards healthy breathing and the training of different respiratory muscles[55]. BreathBalanz is a training tool for your breathing and is not able of collecting breathing data during a day in order to review it later. The BreathBalanz wearable is mounted in the abdominal region using an elastic band and uses an accelerometer to capture breathing data.

Figure 2.4: BreathBalanz[55]

Spire Health Tag

The spire health tag (see figure 2.5) is a health tracker that tracks activity, pulse rate and respiration, in order to capture breathing data the health tag must be worn on the front of the body with tight fitting clothing. The health tags adhere to clothing such as pants or bras and are washer and dryer safe with a battery that lasts for about a year with no charging. The health tag contains an accelerometer, optical heart rate sensor and a proprietary force sensor to measure respiration, which measures the contraction and expansion of the thoracic cavity[56].

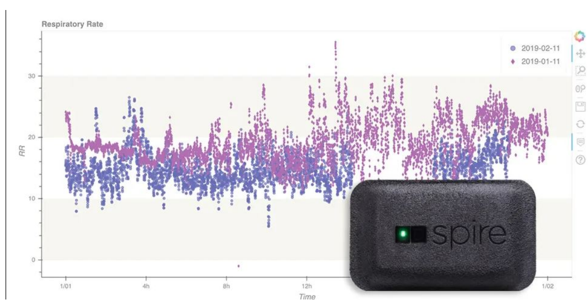

Figure 2.5: Spire Health Tag[56]

Lumo lift

The Lumo Lift (see figure 2.6) posture correction wearable, which is created to help users sit up straight and prevent slouching. The Lumo lift is also able to track steps distance and calories when connected to the application. Daily posture progress can be monitored in the application as well. The Lumo Lift is mounted on the user's shirt with a small magnet right below the user's collarbone. The button is pressed once to set the optimal posture and once the user deviates from this posture the wearable will vibrate in order to notify the user[57]. The Lumo Lift uses a single accelerometer to capture orientation data[58].

Figure 2.6: Lumo Lift[57]

Upright Go 1 & 2

The upright Go (see figure 2.7) is a posture tracker that is attached to the upper back, it gives real time feedback on posture via vibrations in the training mode but can also only be used as a tracker. The upright go was created to improve the posture of its users and the device is connected to your back via a reusable silicon adhesive. In the application an analysis can be found of the posture data which rates the user's daily posture with a percentage[59]. The original Upright GO used a single accelerometer for posture detection, the Upright GO 2 uses both an accelerometer as well as a gyroscope[60].

Figure 2.7: Upright GO 1 and GO 2[59]

Alex+

Alex+ (see figure 2.8) is a wearable designed to prevent text neck, an increasingly common neck injury often related to looking down on a smartphone for extended periods of time. Alex+ is worn over the ears and around the neck and gives vibration feedback when a bad posture is held for a by the user designated period of time. The accompanying application provides feedback on the user's performance and has a coaching program and designated exercises which can be performed[61]. Alex+ uses a single accelerometer to collect orientation data[62].

Figure 2.8: Alex+[61]

Opter Pose

The Opter Pose (see figure 2.9) was created to improve posture, fitness and sleep and can also track ultraviolet exposure and light levels. It can be worn around the neck with a necklace or clipped on to clothing. The Opter Pose gives feedback on posture via vibrations and the connected application keeps track of posture data and all the other functions[63]. It is unknown what type of sensor the Opter Pose uses, however an accelerometer and a light sensor are expected.

Figure 2.9: Opter Pose[63]

8sense

The 8sense (see figure 2.10) wearable is a fitness and body tracker, it captures posture and movement data, gives feedback and recommends activities all based on the user's personal data. The 8sense wearable is worn on the back where it is connected to the user's shirt with a clip, it provides vibration feedback if you are sitting in an incorrect position. The wearable is accompanied by an application which saves the user's statistics and also provides additional feedback partially through notifications. The application coaches the user to be more active and gives suggestions for exercises based on the data it collected[64]. The 8sense wearable uses an accelerometer and a gyroscope to gather data.

Figure 2.10: 8sense[64]

TruPosture

The TruPosture (see figure 2.11) is a smart shirt that incorporates multiple sensors to track the curvature of the spine, when a deviation is detected from the user's set reference posture the shirt gives vibration feedback in the affected area. The shirt can also be used to track and record the user's posture in real time via an application. The main selling point of the TruPosture shirt is that it uses multiple sensors as opposed to the competition which often use only one sensor, the shirt is worn under the user's normal clothing and is waterproof so that it can be washed[65]. The TruPosture shirt uses five accelerometers to get orientation data for five different spots along the spine of the user.

Figure 2.11: TruPosture[65]

2.3.2 Analysis

The other players in the field use either an accelerometer or an IMU to measure posture, however placement of the sensor differs from Breathline. The Lumo Lift and Opter Pose measure posture from the front of the chest, while the Upright GO 1 & 2, Alex+ and 8sense measure posture from the upper back or neck and the Prana measure posture from the belt, from this can be concluded that variety of positions are viable for sensor placement. The TruPosture shirt is unique because it uses multiple of the same sensors to measure posture more accurately. A complete overview of functions, sensors and placement is given in figure 2.1.

	Prana	BreathBalanz	Spire Health Tag	Lumo Lift	Upright GO 1	Upright GO 2	Alex+	Opter Pose	8sense	TruPosture
Breathing	Yes	Yes	Yes	No	No	No	No	No	No	No
Posture	Yes	No	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Activity	Yes	No	Yes	No	No	No	No	Yes	Yes	No
Type of sensor	IMU	Accelerometer	Accelerometer Force sensor	Accelerometer	Accelerometer	IMU	Accelerometer	Unknown	IMU	5x Accelerometer
Placement	Belt	Abdomen	Belt	Chest	Upper back	Upper back	Backside neck	Necklace	Backside neck	Shirt

Table 2.1: Overview of functions, sensors and placement.

2.4 Conclusions for ideation

This section concludes the background research and lays out the decisions made after doing a literature research as well as a market research. This background research has answered the first two research questions and has given us insight in how current implementations function.

It turns out body position and posture are the main two influencing factors on abdominal breathing, and that movement without changing posture does not have an effect on breathing ratio. Abdominal breathing is more prevalent in more horizontal positions such as the supine position, and posture influences how much the belly is able to expand. Because of this the main focus in the ideation and realisation part will lay on detecting these factors and only when they are achieved detection and classification of movement will be focused on. The type of sensor that will be used is an accelerometer as this sensor is already integrated in the Breathline wearable, if during the realisation part of this project it becomes evident that just an accelerometer is not enough to get wanted results a switch to an IMU can be made to be able to achieve the wanted results.

3 METHODS AND TECHNIQUES

This chapter summarises the techniques that will be used during the ideation, specification and realisation phase of this project. The Creative Technology design process is summarised as well as various techniques used in each of the phases.

3.1 Creative Technology design process

The Creative Technology design process (see figure 3.1) is marked by four phases, an ideation, specification, realisation and evaluation phase. Each phase starts and ends with a set of intermediate results, and is defined by a spiral form, meaning that the phase can start and end with any one of the sub phases[66].

3.1.1 Ideation

The ideation phase elaborates on the project idea, sets the problem requirements and generates ideas on experience and interaction. This is achieved through various processes, a stakeholder analysis, brainstorming and ideation sessions. It also deploys tinkering: the process of taking existing technology and identifying new applications for it. The spiral form of this phase is what makes the Creative Technology design process unique, the ideation phase can start of with a piece of technology for which new novel applications are found, bridging the gap between available technology and user needs. Each of these processes uses a diverging and converging stage, in the diverging stage the design space is explored and possible ideas, technologies, experiences and interactions are formulated, in the following converging stage the ideas, technologies, experiences and interactions are defined and the user requirements are formed. Finally, the results of this phase are the conceptual ideas for the prototype which are then evaluated and if found satisfactory used to continue on with the specification phase[66].

3.1.2 Specification

The specification phase is marked by series of prototypes which are used to explore different designs, each design is evaluated and the feedback is used to make changes to or create a new prototype. Added or changed functionalities may influence the user experience which in turn may change their expectations for said functionalities. In this spiral process of rapid prototyping and evaluating user feedback is the driving factor, and prototypes are often reduced to a limited amount of components that represent a single part of the experience to reduce time and cost of prototyping[66]. The result of the specification phase is a defined set of requirements for the final product.

3.1.3 Realisation

The realisation phase is where the product specified in the specification phase is realized through the use of engineering design. First the start of the specification phase is decomposed in order to get an overview of the necessary components to realise the product, then all the components are realised and integrated and finally an evaluation takes place to see in all the user requirements are met. The design model used in the realisation phase will be the waterfall model, which is a linear process allowing for easy backtracking in the case of wrong components or user requirements that are not met[66].

3.1.4 Evaluation

The evaluation phase is all about user testing, and may address multiple aspects. In this phase it is identified if the requirements set in the specification phase are met, this is usually done through user testing. Functional testing can also be part of the evaluation phase which addresses the functional requirements set in the specification phase. As an extra step the results of the project so far can be compared with related work in order to see where it can be placed in the existing context.

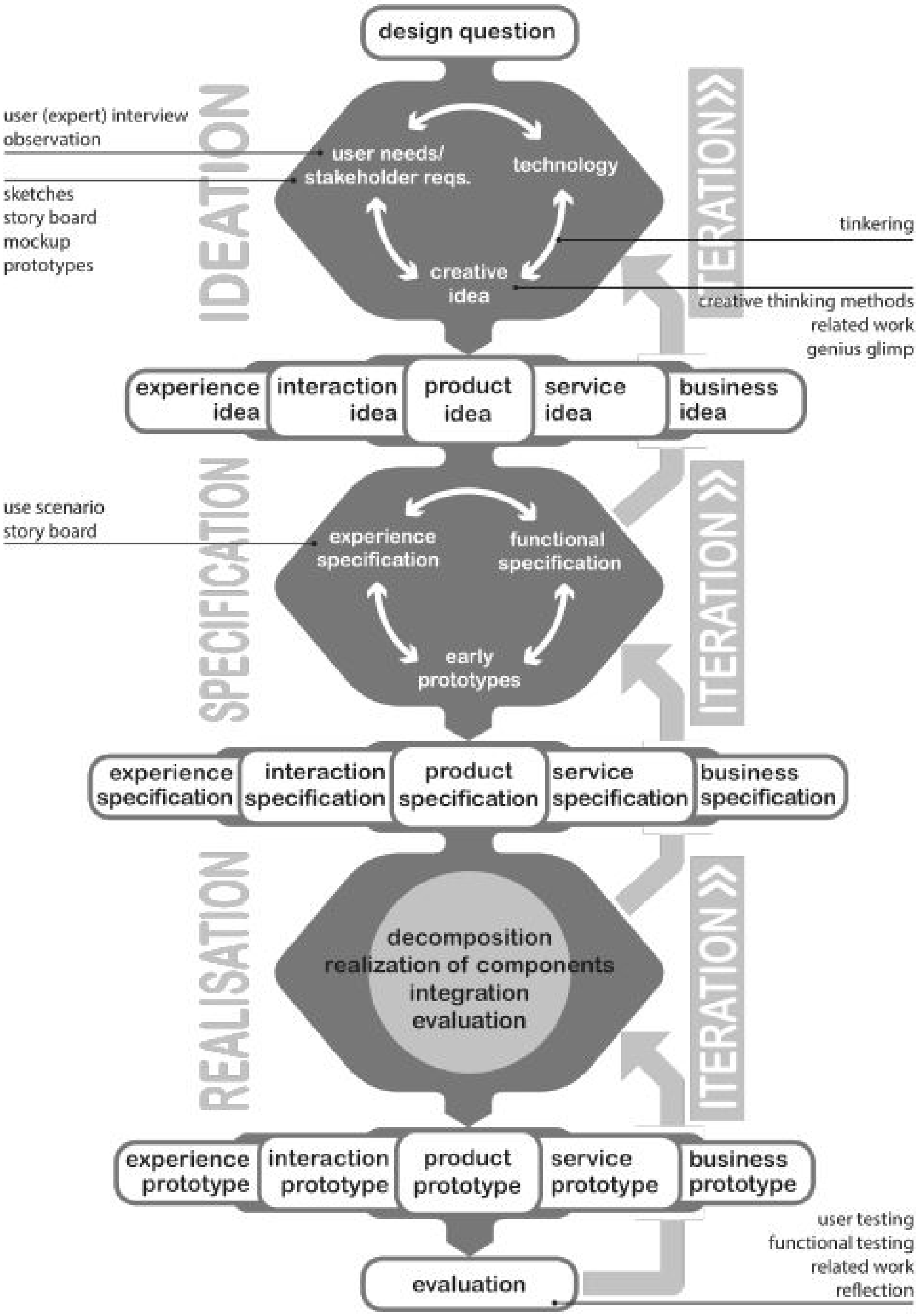

Figure 3.1: The Creative Technology design process[66]

3.2 Stakeholder analysis

The purpose of a stakeholder analysis is to identify all people that have an interest in the project, which is important because it identifies the parties and individuals who have possible influence in the project. When the parties and individuals that have influence in the project are identified they can be ranked on their interest and influence on the project accordingly in table and displayed in a a power grid matrix as described by Bailur[67]. This provides a quick overview of the stakeholders and their relative power. Parties involved in a project usually fall into the following categories: Users, Developers, Legislators and decision makers.

3.3 Brainstorming

Brainstorming is an activity where an effort is made to solve a particular problem, brainstorms can be done in a group format as well as alone. The idea behind brainstorms is to create a free flow of ideas and to not interrupt others with critical points until after the brainstorm session is finished. There are various brainstorm techniques discussed in literature, Bondardel and Didier[68] explain that brainstorming is structured by four rules: generate as many solutions as possible, defer judgement about solutions until the end of the generating sessions, try to come up with original ideas and combine and build on existing ideas.

3.3.1 Rapid ideation

With rapid ideation context is provided to the participating team members with information on the topic and possible limitations. Then a time limit is set for all team members to personally come up with and generate as many ideas as possible through any means available (writing, drawing, cutting...), and afterwards ideas can be discussed and developed further together with the group or individually[69].

3.3.2 Starbursting

Starbursting is a method of brainstorming for understanding new ideas, it focuses on the generation of questions rather than answers or solutions. These questions can then be used to narrow down on for example user groups or specific design aspects, starbursting can be done both alone and in teams. On a piece of paper or a whiteboard a large six-pointed star is drawn with the product idea or challenge in the middle and at each of the points one of the following words: who, what, why where when and how. With each of these words a brainstorm is started and questions are written down but not answered, the focus is at coming up with as many questions as possible. The questions can be answered when the brainstorm is finished and if necessary starbursting can become an iterative process to further explore the answers of the previous session[69].

3.3.3 Online brainstorming

With the current limitations on in person meetings because of the Covid-19 virus online brainstorming is something that is necessary to explore in order to be able to do group brainstorm sessions. Online brainstorming session, sometimes also known as Brain-netting are becoming more and more common and are a viable tool to brainstorm when meeting in person is not possible. There are various programs available that are purpose made for brainstorming, however it can be as simple as a shared Google document or folder[69].

3.4 Interviews

In order to understand the needs of the potential users interviews can be conducted, interviews give insight in what users think about an idea or a product or application. Individual design choices can also be addressed in interviews to get the users feedback and ideas on the implementation of those design choices. Generally there are considered to be three types of interviews: structured interviews, semi-structured interviews and unstructured interviews.

3.4.1 Structured interviews

Structured interviews use an interview protocol and set questions to guide the researcher, this method is advantageous because it helps keeps the interview focused on the goal of the research, and allows for comparison of answers between different interviewees. However it lacks space for the interviewer to continue on specific questions and to probe the user for ideas.

3.4.2 Semi-structured interviews

Semi-structured interviews use a protocol to help guide the researcher and sometimes there can be a set questions, however the researcher is allowed to add questions and probe the user for other ideas and additional details. A semi-structured interview can also be seen as a guided conversation between the researcher and the user.

3.4.3 Unstructured interviews

Unstructured interviews take place with very few, if any, questions, they can be seen as a normal conversation about the research topic at hand. The relatively formless style allows the researcher to establish rapport and comfort with a user and can be useful in situations concerning sensitive topics. The researcher will probe the user to obtain as much in depth and useful information as possible.

3.5 Expert reviews

Expert reviews are a valuable tool for getting quick feedback on the various ideas gathered in the brainstorming sessions and during interviews. During an expert review an expert in the field of that subject will review the work based on relevance, feasibility and practicality such that a collection of viable requirements and implementable ideas is left.

3.6 PACT analysis

A People, Activities, Context and Technology (PACT) analysis is a useful tool when thinking about design, it gives a detailed overview of who will use the product, what they will do with it, in which context and the technology required for that. A PACT analysis clarifies the goal of the project and allows for evaluation on what users want and gives a good starting point for the user requirements. A simple idea for a product or a user scenario can be a starting point for a PACT analysis[70].

3.7 Requirements

User requirements describe what the users require from the product, and thus what the product developers need to focus on. User requirements are written as a validation of what the product wants to deliver and what the users actually need. User requirements should include user-system interaction requirements and use-related quality requirements, and if appropriate recommendations for design solutions that emerged from these user requirements[71]. User-system interaction requirements specify what the product should be able to do, for example in the case of Breathline be able to measure breathing pattern and store this data on the device. Use-related quality requirements are about how the user perceives the product such as effectiveness, efficiency, satisfaction and others[71]. Each of these categories of user requirements will be ranked on the MoSCoW scale, MoSCoW stands for Must have, Should have, Could have, and Won't have. Prioritizing the requirements using a system like MoSCoW is useful because it allows focus on those requirements that are the most important, understanding what work is most important to be done is necessary to make progress and keep deadlines[72]. And will be divided in functional(FR) and non-functional requirements(NFR), functional requirements specify what the system should do and non-functional requirements specify how the system works.

3.7.1 Must have

These are the minimal requirements needed for the product to deliver, these requirements fall in the category if this is not met cancel the project[72]. An examples is: if this project does not come to a solution for classifying posture than their is no need for the product to be developed further.

3.7.2 Should have

Should have requirements are those requirement that are important to the product but not vital, these requirements can be left out if absolutely necessary but rather not. They are differentiated from the Could have requirements by the amount of pain caused in terms of people affected or business value lost[72].

3.7.3 Could have

These requirements are wanted but less important to the usability of the product or the needs of the users. There is less impact in leaving these out but they will be delivered in a best case scenario. They are also the first things to be dropped when the time frame turns out to not be viable[72].

3.7.4 Won't have

These are all the requirements that are not viable for the time frame of this project. They are recorded nonetheless to clarify the full scope of the project. This helps manage expectations by stating that some requirements will simply not make it into the project[72].

3.8 FICS

A Function and events, Interaction and usability issues, Content and structure, and Style and aesthetics analysis (FICS)[73]. Where the PACT analysis takes the point of view from the user when doing a system or product analysis the FICS analysis is a method of analysing a system from the systems point of view. Functions and events is about how the system reacts to user input or other events, Interaction and usability shows what the intended way of using the system is for a user. Content and structure is how the system works on a technical level, how data is gathered, where the data is stored and how it is accessed. Style and aesthetics is a description of the intended look and feel of the system or product.

4 IDEATION

The goal for the ideation phase is to come up with the requirements for the final product as well as to look into what classification methods can be used to classify accelerometer data. This phase will be concluded with a list of requirements which will be taken to the specification phase and by answering the third sub research question: What classification methods are available and which methods are usable on a small microprocessor for classifying a person's posture, movement or position?

4.1 Stakeholder analysis

This project has a variety of stakeholders which will be identified and explained in this section. an overview of all stakeholders and there interest and influence values can be found in table 4.1 and figure 4.1.

4.1.1 Users

General users and patients

The Breathline wearable has a potentially wide user base, if offers functionality in those wishing to train their diaphragmatic breathing via the RIP sensor and posture via the integrated accelerometer. The device can be used by students and people working in an office to achieve correct sitting posture and breathing pattern to optimize their focus and work productivity. People suffering from various breathing related syndromes such as asthma or COPD can use the wearable to train their optimal breathing pattern to prevent symptoms and increase quality of life. The final user group is patients of the client Parvis Sassasian who are in the age group 25 to 55 years old and are interested in knowing about and having better control over their physical and mental health through breathing.

Medical professionals

Medical professionals can use Breathline to monitor patients or give them specific breathing exercises. Breathing and posture data that is gathered can be reviewed by the medical professional and evaluated to detect breathing disorders in early stages. This is possible because of the constant monitoring and combination with body position, movement and posture data.

Researchers

The Breathline wearable offers an opportunity for researchers to track patients for longer periods of time, and find new possible insights in the relation of breathing and posture, body position and movement.

4.1.2 Developers

The Breathline wearable currently has six different developers. Ben Bulsink is the main developer and creator of the Breathline wearable. Tijmen Smit is developing the wireless functionality using Bluetooth, Radhika Kapoor and the author (Martijn Poot) of this project are working on body position, posture and movement dectection making use of the integrated accelerometer and other sensors. Finally the last developpers are Zaccaria Di Giorgio and Dimana Stambolieva who are working on the corporate identity of the Breathline company. Decisions about functionalities and design of the wearable will have to be made together with Ben Bulsink and the other developers.

4.1.3 Legislators

The European Union is currently in a transitional phase from the Medical Devices Directive (MDD) to the Medical Devices Regulation (MDR) which will go into full effect on may 26th 2021 (delayed from 26th may 2020 because of covid-19), under the MDD wearable devices that monitor physical processes fall into class I which means that that are not subject to the regulations of medical devices. Under the new MDR wearables that are intended for monitoring of physiological processes are classified as class IIa or IIb which means that they need to be certified by a notified body[74]. The TÜV[75] is one of the largest corporations that certify medical devices and as such could be a stakeholder if the Breathline wearable needs to be certified.

4.1.4 Decision makers

There are multiple decision makers for this project. At first there are the clients of this project Ben Bulsink and Parviz Sassanian. Their visions and goals will be crucial for the course of this project.
Then there are the supervisor Erik Faber and critical observer Cora Salm who will provide feedback and support during this project, oversee its progress and set the time frame for the project. Finally they will grade this project based on the quality and structure of this report, the quality of the work, the process of the graduation project and the public presentation and defence.
Finally the other developers as mentioned in section 4.1.2 in this project are also decision makers, as their decisions can limit or enable features of this project.

Stakeholder	**Category**	**Interest**	**Influence**
Adults	User	Low	Low
Patients (Breathing syndrome)	User	Medium	Low
Patients (Parviz Sassasian)	User	Medium	Low
Medical professionals	User	Medium	Low
Researchers	User	Medium	Low
Ben Bulsink	Developer/Decision maker	High	High
Parvis Sassasian	Decision maker	High	High
Erik Faber	Decision maker	High	High
Cora Salm	Decision maker	Medium	High
Radhika kapoor	Developer/Decision maker	Medium	Medium
Tijmen Smit	Developer/Decision maker	Low	Medium
Zaccaria Di Giorgio	Developer/Decision maker	Low	Low
Dimana Stambolieva	Developer/Decision maker	Low	Low
TÜV	Legislator	Low	High

Table 4.1: Interest and influence of stakeholders

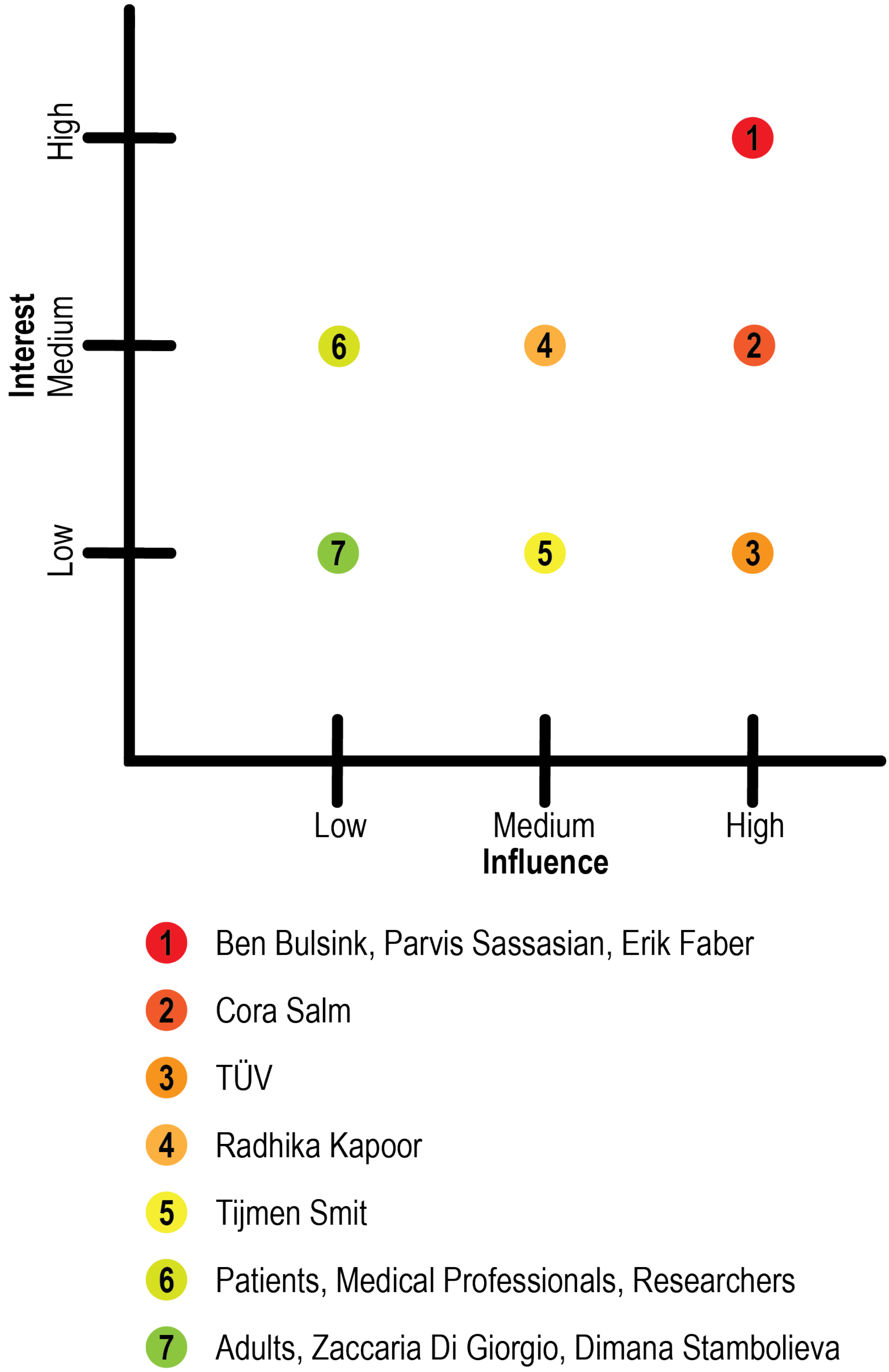

Figure 4.1: Matrix of interest and influence of stakeholders

4.2 Individual brainstorming

Individual brainstorms on several topics were conducted, a starbursting session was done to generate questions about possible user requirements, another session was done to generate ideas for data collection methods, and finally, a starbursting session was done to create an overview of data classification methods. The results of these brainstorms can be found in the sections 4.2.1, 4.2.2 and 4.2.3.

4.2.1 User requirements

The results from the starbursting session on user requirements is given below. These questions will be answered and discussed in section 4.4.

- Who
  - Who is working on this product?
- What
  - What information does the user want to know?
  - What interaction does the user have with the device?
  - What battery life is required by the user?
  - What method of feedback should Breathline have to notify user of bad posture?
  - What should the device look like?
  - What benefits does the device offer to a user?
- Why
  - Why would a user choose Breathline?
- Where
  - Where would users use the device?
  - Where would users prefer the device to be mounted on their body?
- When
  - When is it necessary to calibrate the device?
  - When does a user need to connect the device?
  - When would you use the device?
  - When would users use the device?
  - When is the best moment to notify the user of bad posture?
- How
  - How would disabled users interact with the device?
  - How do you calibrate the device?
  - How do you turn on or of the device?
  - How do you charge the device?
  - How to turn on or of posture sensing?
  - How would Breathline influence your work

4.2.2 Data collection

The results from the starbursting session on data collection is given below. These questions will be answered and discussed in section 4.4.

- What
  - What sensors are available?
  - What are the power restrictions?
  - What would be the ideal mounting location for posture data?
  - What would be the ideal mounting location for body position data?
  - What modifications need to be made to the Breathline in order to accommodate the chosen sensor?
- Why
  - Why is it necessary to have calibration?
- Where
  - Where will the sensor be located in the device?
  - Where is the data stored?
- When
  - When should data be collected?
  - When should the sensors be calibrated?
- How
  - How can the sensors be calibrated?
  - How many degrees of accuracy should the sensor have?

4.2.3 Data classification

The results from the starbursting session on data classification is given below. These questions will be answered and discussed in section 4.4.

- Who
  - Who would benefit from data classification?
- What
  - What methods of data classification are there?
  - What are methods of data classification for accelerometer data?
  - What are methods of data classification for IMUs?
  - What categories should the sensor data be classified in?
  - What categories of posture should there be?

- Why
  - Why is it necessary to classify data?
  - Why analyze both breathing and posture, body position, and movement data?
- Where
  - Where is the data classified?
- When
  - When is it preferable to use machine learning over parameters?
- How
  - How do classifiers work?

4.3 Group brainstorming

A group brainstorm was executed with two students, student one was a student of molecular science and technology and student two studies business economics, both students are both physically active and have used sports trackers in the past. The brainstorm was executed using Google documents and Discord with the purpose of drafting up features that are expected of the Breathline wearable and a user testing protocol for testing the prototype wearable. The participants were not acquainted with abdominal breathing so a short explanation was given of what abdominal breathing is and what benefits it might offer. Both participants were aware of what bad postures look like and the possible symptoms of bad posture.

4.3.1 Features

The focus for this part of the brainstorm was on features regarding posture, body position and movement. Both participants were interested in a feature that would inform the user of bad posture, suggested was either a sound or a vibration when the user is not conforming to correct posture, "as long as it is as annoying as possible" was suggested by one of the participants. One of the participants came up with the idea of some sort of posture score which represents the amount of time the users posture is correct. This score would be influenced by longer periods of time spent in an incorrect posture and by the amount of times the posture correction is ignored.

One of the participants explained that he had a bad habit of not using some products he bought in the past because they were fiddly and required user input too much. He suggested that the wearable should be put on and forget about it as much as possible. A system that is able to remember one's perfect posture and not having to reset it every time you want to use it. Finally, a discussion about wireless app functionality took place and the benefits of such functionality were acknowledged. However, that is outside of the scope of this project and therefore not included.

An overview of the features in short:

- Feedback when the users posture is incorrect.
- Posture score.
- Small size and no buttons.
- Automatic calibration or "remembering" of posture.

4.3.2 Experimental protocol

Because of the implications of the current covid-19 crisis the participants that will be recruited will be roommates of the researcher. The participants will be introduced to the subject and asked to put on the prototype wearable and wear it for a maximum of one hour, during this hour the researcher will note the average posture, body position or movement for each minute for a total of 60 measurements. After this period of time the participant can be asked questions about the device.

The participants of this brainstorm suggested to have the participants of the user testing perform a set of standard exercises in order to get at least a one minute measurement of each type of posture, body position and movement. As well as having data available for possible machine learning data classification.

4.4 Preliminary concept

4.4.1 Product features

The product features can be extracted from the individual starbursting brainstorms, the group brainstorms, interviews and the literature research that was executed. A list of possible and required features is given.

- The user should receive real time feedback on his/her sitting posture.
- The user should be able to view his/her data afterwards.
- The user should be able to calibrate the posture tracker.
- The user should be able to turn the device on and off.
- The user should be able to use the device for at least a whole day.
- The device should be comfortable to wear for the user.
- The user should be able to turn the posture and body position sensing functionality off.

4.4.2 Data collection

There are various methods and tools available that can work for posture, body position and movement data collection, and generally in literature two methods are described. Method one is via computer vision, stationary camera's track a participant who usually wears brightly colored tags to improve tracking. Method two is using MEMS sensors such as accelerometers and gyroscopes placed on different parts of the body. Then there are various novel methods such as the system created by Huang et al.[76] which uses a receiver tags placed on the participants body and a ultra wide band radio system which is able to calculate the distance to each tag. Other potential sensors can be imagined such as strain sensors integrated in clothing [77], or pressure sensors integrated in the chair of the users[78].

Breathline is a wearable product. Because of this, small and low power sensors are favored over large room scale methods. Strain sensors are a low power option, which require, however, either extra mounting locations or integration into clothing. Strain sensors are also a very new approach not yet being used by competitors on the market and having not much success documented in literature. Accelerometers and gyroscopes have been well documented in literature (see section 2.2) and are used by all competitors successfully (see section 2.3). The abdominal mounting location of the Breathline is different then competitors which show an ideal location for posture sensing to be the upper chest or back.

It is necessary for the device to have some sort of calibration, either manual, automatic or a combination of both. This will allow the user to calibrate the device to the ideal posture every time it is used, giving a baseline for the posture and body position sensing. Data should be collected with a frequency of four or eight Hertz as these frequencies are already being used by the Breathline wearable. The sensor should at least be accurate enough to detect a posture change of ten degrees at the upper chest and back with respect to the upright position.

- The product should collect data using an Arduino.
- The product should make use of an accelerometer and if necessary a gyroscope.
- The product should be mounted in the abdominal region.
- The product should collect data with a frequency of four or eight Hertz.
- The product should be able to detect a ten degree change of posture in the upper chest and back.

4.4.3 Data classification

Categories

The data collected by the sensors of the product will need to be classified in various categories, starting with the base categories posture, body position and movement. An overview of categories is given below:

- Posture
  - $> 10^\circ$ forward posture
  - upright posture
  - $> 10^\circ$ backward posture
- Body position
  - Upright (sitting or standing)
  - Laying down
- Movement
  - Walking

Classification methods

In order to place the collected data in a certain category it has to be classified either with the help of a classifier or by hand. There are various types of classifiers, which can usually be categorized in either linear or non linear classifiers. Linear classifiers can be used when categories can be separated by a single line in a 2d space as can be seen in figure 4.2, non linear classifiers can be used when the category boundaries cannot be approximated well with linear

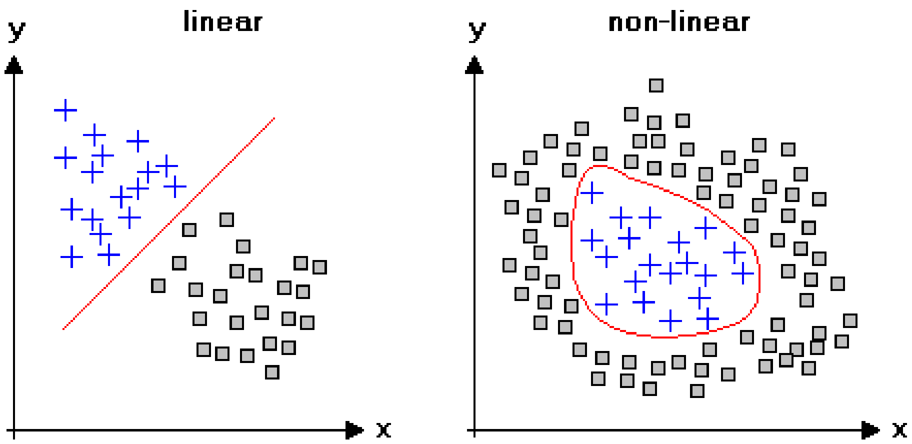

Figure 4.2: Example of a linear and a non-linear classifier in a 2d plane[79]

hyperplanes. Non-linear classifiers will be more accurate in such applications. However, they require more computing power. An example of a non-linear classifier can be found in figure 4.2. For the classification of accelerometer data non-linear classifiers such as support vector machines(SVM) or artificial neural networks(ANN) and k-nearest neighbour(k-NN) are the most logical choice. SVM and ANN classifiers work through supervised learning, ANN makes guesses on what the distribution in the categories might be and then needs to be corrected in order for the neural network to learn and adjust it distribution weights accordingly. SVM in short uses a labelled training data set and then transforms the input data into a high dimensional feature space in which it can find a linear relationship, it basically takes input data and assigns it to either one of two categories. k-NN is a non-parametric and lazy learning algorithm, it simply matches and categorises a new data point to the majority of the nearest neighbours. where the amount of neighbours looked at is determined by K (figure 4.3). k-NN has been successfully used for activity recognition using an accelerometer by Kaghyan et al.[80].

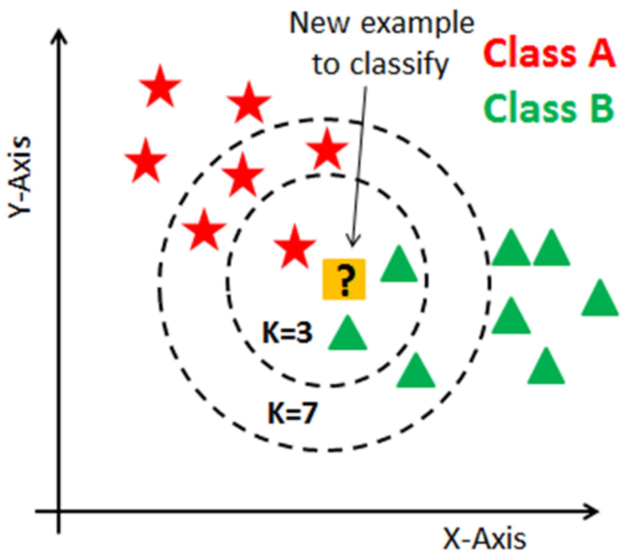

Figure 4.3: Example of a k-NN classifier in a 2d plane[81]

4.5 PACT analysis

A PACT analysis as explained in section 3.6 is executed, it gives an explanation of the system from a users point of view.

4.5.1 People

The people that will use Breathline are a very diverse group, from novices who are looking for a product to try out breathing exercises to the health conscious individual who is looking for a product to track their breathing behavior for the whole day and get insights in their performance. Two personas are drawn up to get an insight into a possible user.

Manon is a 45 year old secretary who works in a law firm. She spends long days at the office mostly sitting. She read online about the effects of breathing therapy on stress reduction and wants to try these techniques to see if they can help her focus better and reduce stress while working. She downloaded a breathing therapy app with exercises which she has used for a little while now. However this app is not able to show her actual breathing and because of that she is not sure if she is applying correct breathing techniques and whether her posture might influence her breathing pattern. Because of these reasons she is looking for a product that will give her insight in her breathing patters and sitting posture both while doing exercises as well as at work.

Hans is 32 year old researcher at the University of Twente at the faculty of Engineering Technology in the field of industrial design. While in his field he is mainly focused on the design of consumer products. In his spare time he likes to play around with wearable health products. In the past he has used several breathing related wearables, all making use of accelerometers to capture his breathing pattern. Not convinced of their performance he is looking for a wearable which uses a more reliable technique and which he can not only use during breathing exercises but also during his regular work day. He expects Breathline to capture breathing and posture data and give him feedback on his breathing and posture performance both in real time and on request.

4.5.2 Activities

The Breathline sensor can be used on a daily basis as a monitoring and training tool for abdominal breathing, posture, body position and movement. The Breathline sensor automatically starts monitoring when the belt is connected and can be used actively by doing breathing exercises with the device connected to a computer or passively in which case the device is standalone and the data is loaded from the device later on. Wireless functionality is currently being integrated which will allow for wireless use with either a computer or a phone application.

4.5.3 Context

The Breathline can be used as a monitoring tool during whole day or during more specific moments such as while working, while training or when doing breathing exercises. For example breathing ratio can change during stressful situations and user can use the Breathline wearable to monitor and train their breathing in these situations. The Breathline wearable can be worn on the users abdomen, and can be worn while wearing thin clothes underneath as well as clothes covering the device.

4.5.4 Technologies

Input

The Breathline wearable uses RIP (section 1.2.2) in combination with an Arduino microprocessor to collect the breathing data, and an accelerometer to collect movement or angular data. This data can be analysed using a computer program when the wearable is connected to a computer using a USB micro cable. In order to conduct testing a secondary Arduino is used in combination with an IMU.

Output

The data gathered by the Breathline wearable can be viewed in real time using the Breathline computer application. This application displays the abdominal breathing pattern that is collected by the RIP sensor and the output of the three axis of the accelerometer. The Breathline application is also able to offload any data that is gathered on the wearable to a CSV file.
The prototype that is created for testing will output the accelerometer data and gyroscope data to the computer so it can be viewed in real time as well as saving it to a CSV file on a SD card which can be read out using a computer and a program such as Microsoft excel.

4.6 Conclusion and preliminary requirements

4.6.1 Conclusion

The prototype system that will be created will need to use an automatic classifier for the accelerometer data, this is advantageous because it allows both real time applications such as as a posture correction device as well as feedback when doing breathing exercises. A k-NN classifier can be implemented on a microprocessor with a personalized set of data for each user and has been used successfully in the past to classify accelerometer data.

4.6.2 Preliminary requirements

A final list of requirements to be taken to the specification phase is made and organized according to the MoSCoW method, functional requirements(FR) and non-functional requirements(NFR) are labeled as such.

- Must have
  - FR1: The system must be able to differentiate between the being upright and laying down body positions.
  - FR2: The system must be able to detect walking.
  - FR3: The system must be able to detect forward posture, normal posture and backward posture.
  - FR4: The system must use an Arduino.
  - FR5: The system must use an accelerometer.
  - FR6: The system must be mounted in the abdominal region.
  - FR7: The system must have a calibration option.
- Should have
  - FR8: The system should collect data with a frequency of four or eight Hertz.
  - FR9: The system should use a k-Nearest Neighbour classifier.
  - FR10: The system should allow users to view their data.
  - FR11: The system should be able to be run for a whole day without user interference.
  - FR12: The system should have the option to be turned off by the user.
  - FR13: The system should be able to detect a ten degree change of posture in the upper chest and back
- Could have
  - FR14: The system could give real time feedback to the users on their sitting posture.
  - FR15: The system could have a gyroscope.
  - FR16: The system could have an option to turn the posture and body position sensing functionality off
  - NFR1: The system should be comfortable for a user to wear.
  - NFR2: The system could have a posture score or rating for the user.
- Won't have
  - FR16: The system won't have a Artificial Neural Network classifier.
  - FR17: The system won't have integration with the Breathline wearable.

5 SPECIFICATION

The goal of the specification phase is to work out the idea of the Breathline posture, body position and movement tracker and get a better overview of the functionalities and system architecture. Which will result in a refined set of requirements which are taken to the realisation phase to create the functional prototype.

5.1 FICS analysis

A FICS analysis as described in section 3.8 is executed to get an overview of everything that needs to be included in a functional prototype. Each section will give an overview of processes that happen.

5.1.1 Functions and events

The system will have several functions aside from the already existing functions that cover the breathing data collection.

The system will have the following functions

1. Start up the device when the user connects the RIP band.
2. Collect the raw accelerometer data.
3. Categorize the data in the categories from section 4.4.3
4. Save the raw and categorized data to a CSV file
5. Give real time feedback to the user about their sitting posture.
6. When connected to a computer download the data for further analysis.

The system will capture additional data which can be added to the data captured by the Breathline wearable, this data can then be used to find correlations between posture, body position and movement and the abdominal breathing pattern. If FR13 is implemented it will give feedback to the user about their posture while wearing the device.

5.1.2 Interaction and usability

The user interacts with the system by putting the wearable on, after which the system automatically starts collecting data, and taking the wearable off, after which the system stops collecting data. Additional interaction takes place through the calibration button which the user can press when the automatic calibration is not correct.

5.1.3 Content and structure

Input

The Breathline wearable uses RIP (section 1.2.2) in combination with an Arduino pro-micro microprocessor to collect the breathing data, and an accelerometer to collect movement or angular data. The data is collected at either 4Hz or 8Hz sample frequencies and saved to the Arduino pro-micro, this data can be removed and analysed using a computer program when the wearable is connected to a computer using a USB micro cable. In order to conduct testing a secondary Arduino nano or arduino UNO is used in combination with either a sparkfun LSM9DS1 IMU or a MPU6050 IMU. This device is used while wired to a computer and provides a continuous data stream.

Output

The data gathered by the Breathline wearable can be output in real time using the Breathline computer application, this application displays the abdominal breathing pattern that is collected by the RIP sensor and the output of the three axis of the accelerometer. The Breathline application is also able to offload any data that is gathered on the wearable to a CSV file.
The prototype that is created for testing will output the accelerometer data and gyroscope data to the computer so it can be viewed in real time as well as saving it to a CSV file on a SD card which can be read out using a computer and a program such as Microsoft excel.

5.1.4 Style and aesthetics

The Breathline wearable is mounted in a small white 3d printed enclosure with button snaps to connect the RIP band (figure: 5.1). The functional prototype for testing the posture, body position and movement functionalities will not have a specific design made for it.

Figure 5.1: Breathline wearable

5.2 System architecture

This section will give an overview of the system architecture, it gives an overview of the components in the system and how they interact with each other. The system architecture is described in three distinct levels, level 0 gives an overview of the inputs and outputs of the system, level 1 will give an overview of the whole system including its inner workings. Level 2 gives a more in depth look into the k-NN classifier The system described here is the functional prototype as it is being created, consisting of an Arduino Nano, MPU 6050 IMU, MicroSD card adapter, dS3231 real time clock and a vibration motor.

5.2.1 Level 0

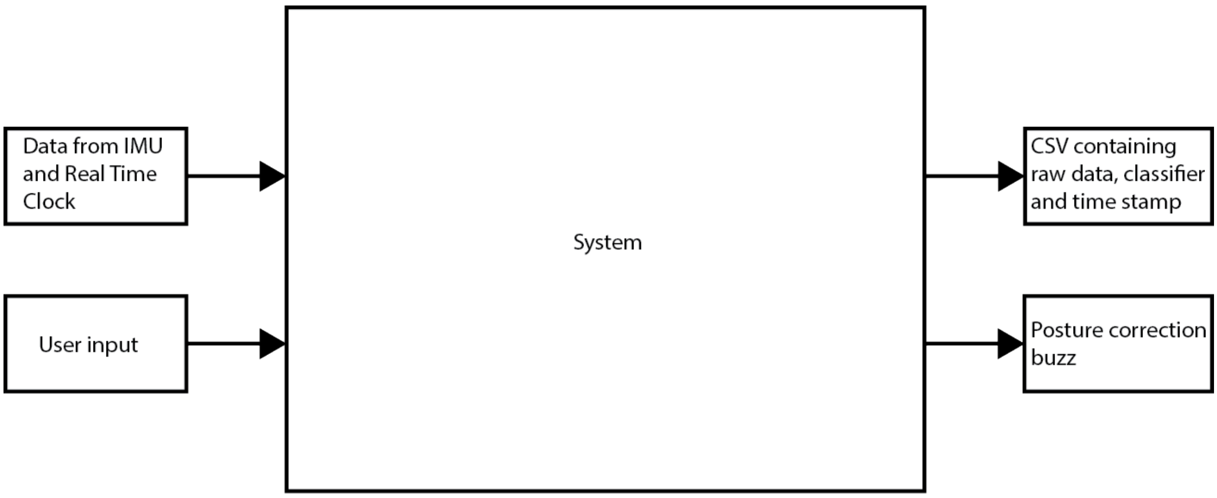

Figure 5.2: System architecture level 0

Level 0 consists of the different input devices and the final output and can be seen in figure 5.2. The inputs of the system are the Accelerometer and Gyroscope data collected by the MPU6050 IMU, the current time and date from the real time clock and the user input in the form of a button in order to calibrate the device. The first output of the system is a CSV file containing the raw accelerometer and gyroscope data, the classification given to it by the classifier and a time stamp, the second output is posture feedback to the user in the form of a buzz if functional requirement 13 is achieved.

5.2.2 Level 1

As a more detailed explanation of what happens inside the system level 1 has been created, level 1 can be found in figure 5.3.

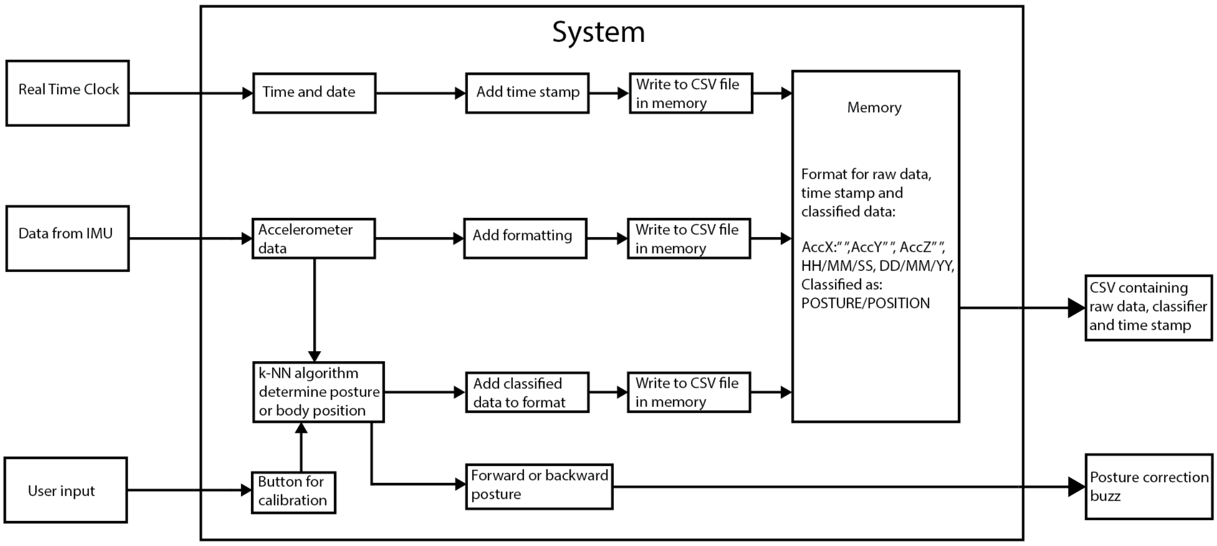

Figure 5.3: System architecture level 1

This level 1 architecture explains what is happening when a user is using the device. It explains how and in what format the data is saved and what the functionalities of the device would be if FR1 - FR13 are met.

On the left side the same inputs as shown in level 0 can be seen. Only the accelerometer data is retrieved from the IMU. The accelerometer data is then copied and used by the k-Nearest Neighbour(k-NN) algorithm to determine the posture or body position according to the categories defined in section 4.4.3. The original data is formatted for readability then a time stamp and the result of the k-NN algorithm is added and the data is saved to a CSV file on a SD card which can be removed and plugged in to the computer to be read.

When the k-NN algorithm detects a minute average of either forward or backward posture the buzzer is used to notify the user of a bad posture. The reason for choosing the minute average is to prevent the buzzer from falsely notifying the user when he or she moves around on their chair.

5.3 Level 2

A k-Nearest Neighbour(k-NN) is one of the simplest ways of classifying data, a new entry is compared with an existing data set and classified according to its nearest neighbours, and the amount of neighbours looked at is determined by k. As an example if there is a new data point and the k-NN classifier looks at k=5 neighbours, and from the five closest neighbours three are classified as normal posture and two as forward posture the new data point will be classified as normal posture. The level 2 architecture visually explaining this process can be found in figure 5.4.

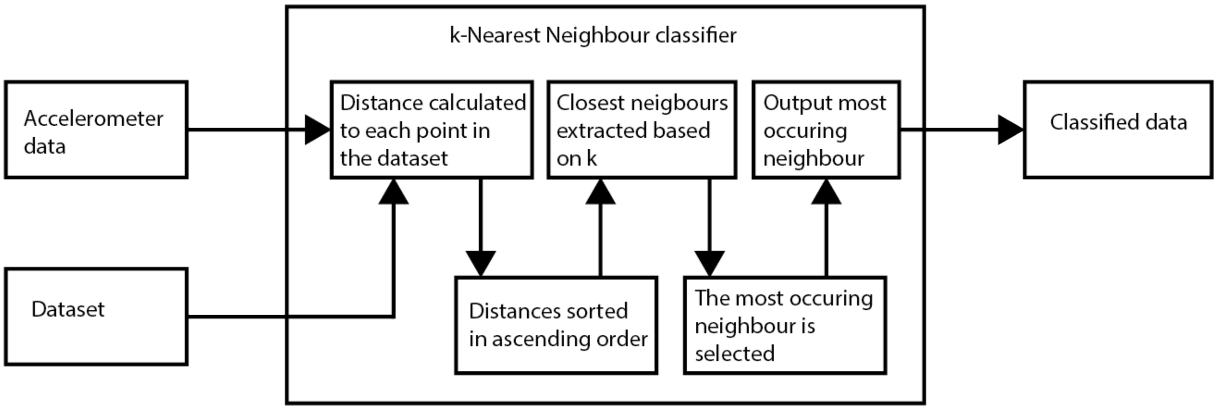

Figure 5.4: System architecture level 2

A visualisation of how the accelerometer data will be classified is given in figure 5.5, green indicates a normal posture, yellow a forward posture, orange a backwards posture, blue indicates lying down and grey walking.

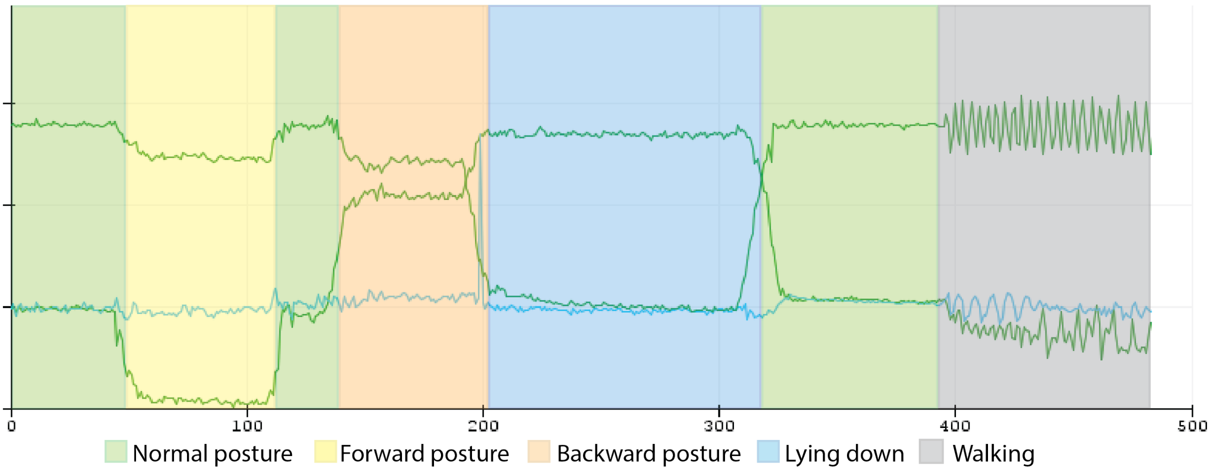

Figure 5.5: Classification example

5.4 Final requirements

A list of preliminary requirements was given in section 4.6, this section will elaborate on the requirements from that section and add in new requirements that followed out of the system architecture, software and hardware analysis. The requirements will follow the same structure as in section 4.6 making use of the MoSCoW method and functional(FR) and non-functional(NFR) requirements.

- Must have
  - FR1: The system must be able to differentiate between the being upright and laying down body positions.
  - FR2: The system must be able to detect forward posture, normal posture and backward posture.
  - FR3: The system must be able to detect walking.
  - FR4: The system must store the data gathered in a CSV file.
  - FR5: The system must use an Arduino in combination with an accelerometer.
  - FR6: The system must be mounted in the abdominal region.
  - FR7: The system must automatically classify the accelerometer data with at least an 80% accuracy.
  - FR8: The system must have a manual posture/body position calibration.
- Should have
  - FR9: The system should have an automatic posture/body position calibration.
  - FR10: The system should collect data with a frequency of four or eight Hertz.
  - FR11: The system should use a k-Nearest Neighbour classifier for real time classifying of data.
  - FR12: The system should allow users to view their raw and classified data.
  - FR13: The system should be able to be run for a whole day without user interference.
  - FR14: The system should be able to detect a ten degree change of posture in the upper chest and back.
- Could have
  - FR15: The system could give real time feedback to the users on their sitting posture.
  - FR16: The system could have an option to turn the posture and body position sensing functionality off, while keeping the breathing data collection on.
  - NFR1: The system should be comfortable for a user to wear.
- Won't have
  - FR17: The system won't have an Artificial Neural Network classifier.
  - FR18: The system won't have integration with the Breathline wearable.
  - FR19: The system won't have a gyroscope.
  - NFR2: The system won't have a posture score or rating for the user.

6 REALISATION

This chapter will describe the process of the development and realisation of the prototype posture wearable, the code that runs on the wearable including the k-Nearest Neighbour classifier. First the development process used to develop the final prototype is laid out, then an in depth explanation is given on the hardware and software used. Finally, the initial testing procedure is described.

6.1 Development process

During the realisation phase the prototype as described in the specification phase has been created through various different Lo-Fi prototypes, these Lo-Fi prototypes were created to test parts of the final Hi-Fi prototype. All Lo-Fi prototypes and the Hi-Fi prototype were tested by the author to confirm that the prototype was working correctly and reached the desired goals.

6.1.1 Lo-Fi prototypes

Lo-Fi prototypes were created to test different parts of the Hi-Fi prototype, the first Lo-Fi prototype was created to test the functionalities of the IMU used in the final prototype and to evaluate whether it is possible to use a single accelerometer to match the requirements set up in the specification phase, this Lo-Fi prototype consisted of an Arduino Nano every and a MPU6050 IMU and the code which can be found in appendix A, this prototype allowed for the visual readout of accelerometer data and verified that it is possible to detect posture using an accelerometer.

The second Lo-Fi prototype consisted of the same hardware but the k-NN classifier was integrated in the code (appendix B), this classifier was coded in C++ and is used as an Arduino library it uses a database stored on the Arduino as the neighbours. The code for that was used for this prototype will be explained more in section 6.4. The purpose of this Lo-Fi prototype was to evaluate whether a k-NN classifier was able to classify accelerometer data. And it was found that a k-NN classifier shows potential in classifying accelerometer data however it is a calculation intensive method of classifying accelerometer data.

The third Lo-Fi prototype was created to test the saving of data to a CSV file on a SD card, this card can then be removed from the prototype and inserted into a computer in order to be read out or used for further classification. The prototype consisted of an Arduino nano every and a unbranded microSD card adapter, example code supplied by Arduino was used as a starting point and adjusted to check the specific needs for this project (appendix C). This proved that it is possible to save a data stream to a CSV file which can in turn be read out by a computer.

6.2 Hi-Fi prototype

A Hi-Fi protype was created to conduct user testing, this prototype consisted of the hardware from the Lo-Fi prototypes, an added real time clock to keep track of the time for the purpose of logging the data and a small vibration motor to give feedback to the user. All of the code from the Lo-Fi prototypes was integrated and code was added for the real time clock and the vibration motor. This prototype was developed to be used in the user testing for the evaluation phase, and was tested by the author to make sure all functions worked properly before continuing to the user testing.

Two changes were made to the original system schematic. These changes are made because of findings during the creating of the Lo-Fi prototypes and are in the user interaction with the device, and the way feedback is given to the user. The new system architecture can be found in figure 6.1.

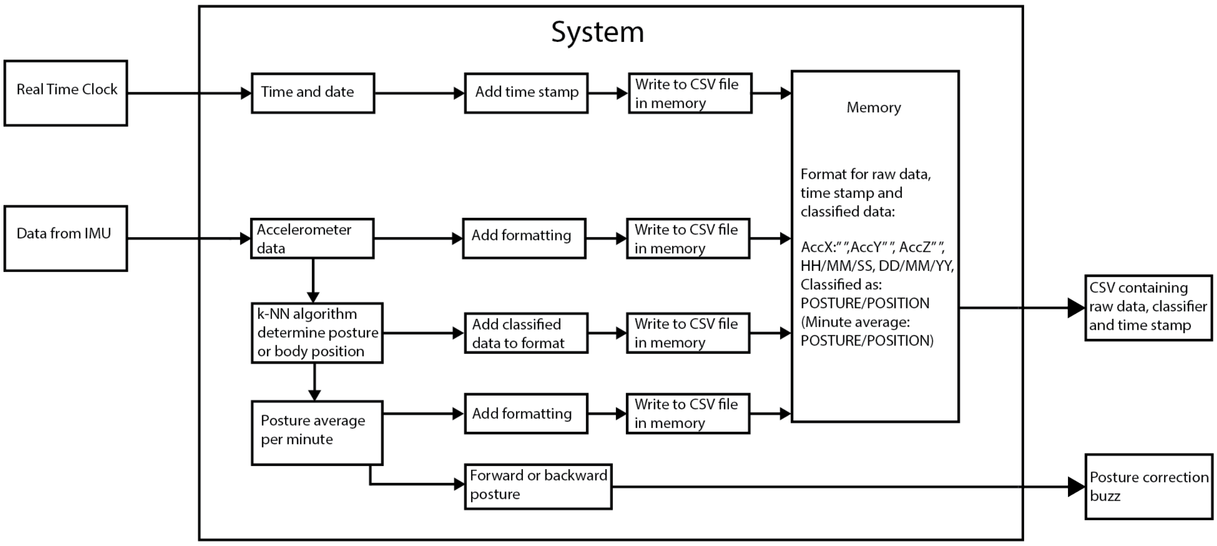

Figure 6.1: Hi-Fi prototype level 1 system architecture

The user input to calibrate the system was removed, because by making use of a k-NN classifier the data set that is used for classifying can be tuned to each user manually removing the need for an option to calibrate the accelerometer position. Additionally writing code so that the user could collect their own data set, and "calibrate" the device was deemed to be unnecessary for the relative small amount of user test that are going to be executed.

Second, it was determined that it is better go give the user feedback based on a one minute average posture, so an array is used to store each classified data point and each minute the most occurring category is written to the CSV file and if this average is either forward or backward posture feedback is given to the user.

A real time clock was added to the system to be able to add a time stamp to the saved data, this is necessary in order to be able to compare the data gathered by the Hi-Fi prototype with the breathing data gathered by the Breathline wearable. This real time clock provides the Arduino with the current time even when disconnected from a computer.

6.3 Hardware

This section covers all the hardware used in the final Hi-Fi prototype. Amongst others a wiring diagram is given which can be used to recreate the wearable, the wiring diagram can be found in figure 6.2.

6.3.1 Arduino Nano Every

The Arduino Nano Every[82] is a small microprocessor board that can be programmed using the Arduino language. It has an upgraded ATMega4809 processor compared to the previous version of the Arduino Nano which clocks in at 20MHz, 48KB of flash memory and 6KB of SRAM. This makes it perfect for the system as the storage requirements for the database are higher then previous versions of the Arduino Nano can offer. The higher processor speeds reduce the time the calculations of the neighbours take as well.

6.3.2 Inertial Measurement Unit

The GY-521 breakout board contains a MPU6050 6DOF IMU[83], a combination of an accelerometer and an IMU for this project only the accelerometer data is used. The GY-521 was chosen because of it cheap price, availability and the integrated Digital Motion Processor (DMP) which fuses the accelerometer and gyroscope data together to minimize errors and reduce gyroscope drift. For this project only the accelerometer data is used and because of that the DMP functionalities are not used.

6.3.3 MicroSD Card Adapter

An unbranded MicroSD Card Adapter[84] is used in order to store the captured data and the classification of that data on a CSV file, this MicroSD card can then be removed from the wearable and inserted into a computer for further inspection.

6.3.4 Real Time Clock

A RTC DS3231 real time clock (RTC)[85] is used to provide the data points with a time stamp. The RTC has its own 3V battery in order to keep track even when the Arduino is disconnected from power. The RTC can keep track of hours, minutes and seconds as well as days, months and years.

6.3.5 Vibration motor

A small vibration motor in combination with a mosfet is used to give the user feedback on their posture. It is connected to the 5V output and the voltage is scaled down by making used of the mosfet and the Arduino code.

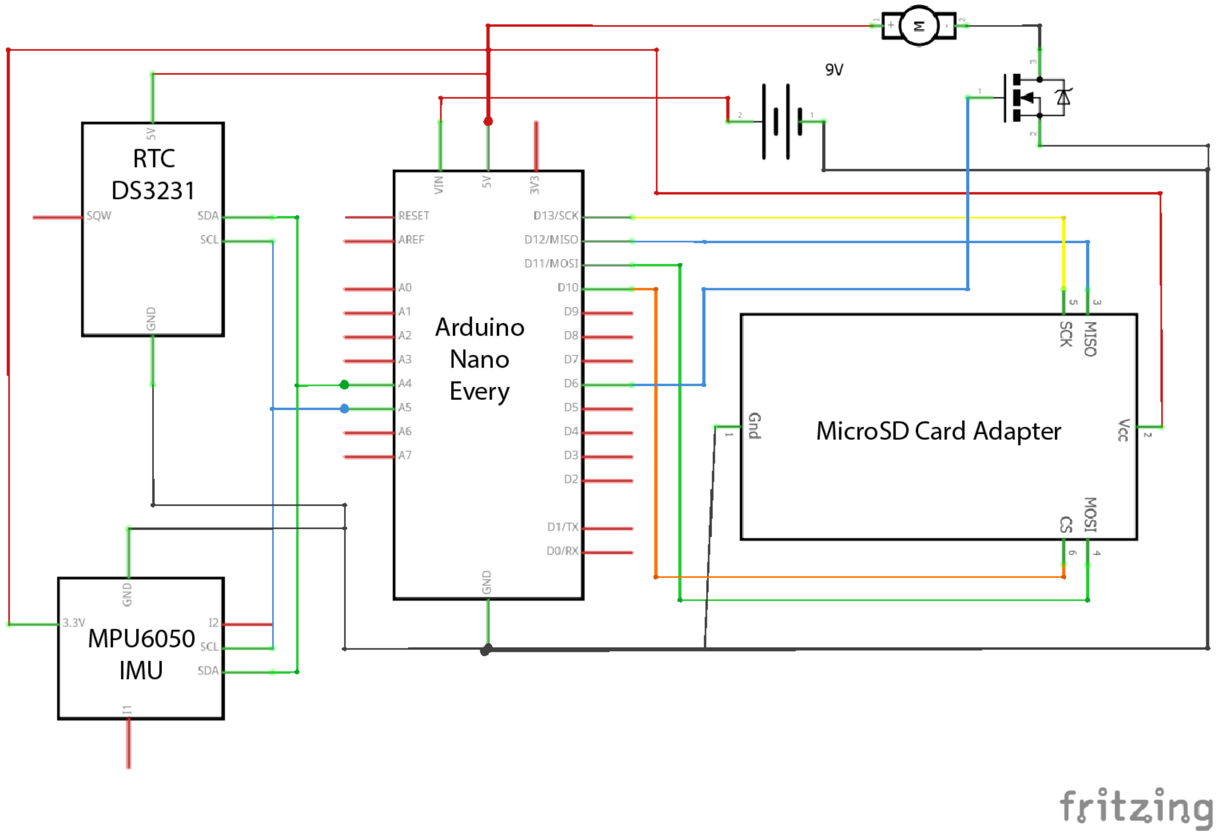

Figure 6.2: Wiring diagram for the final wearable, created using Fritzing[86]

6.4 Software

This section covers the software that runs on the wearable. First a brief explanation on the used languages and program's is given, then an explanation of the code is given in the form of a walk-through.

6.4.1 Arduino

The Arduino language and the Arduino Integrated Development Environment (IDE) application are used to write the main code for this project. The Arduino language is C++ with its own set of code structure rules. The Arduino IDE application allows for both writing and uploading of the code to the Arduino micro processors. The Arduino environment was chosen because of familiarity with the language and because the Breathline wearable already uses an Arduino, so in the future the integration of both wearables should not be difficult.

6.4.2 C++

The second programming language used is C++, C++ is a general-purpose programming language with lots of applications. Libraries that can be used by the Arduino language are written in C++ and for this project an Arduino library was written making use of the Arduino IDE editor and the Microsoft visual studio editor. The decision was made to write the k-NN classifier in C++ as it allowed for a more easy integration of a separate database and a considerable cleanup of the main program.

6.4.3 Code explanation

The code for the prototype that will be used for user testing can be found in appendix D. A walk through of the functions of the code will be given here, The code will be explained in two sections, what happens when the device is started up and what happens when the device is running.

Startup

When the wearable is connected to a power source a startup sequence is initiated, this startup sequence consists of the following parts. First the connection with the MPU6050 is started up and the device is awakened, then the program checks if there is an SD card inserted, if this is not the case then the program does not continue further and sends an error message. After that the real time clock is started up and a one minute timer is set, following this the time is used to write a starting message to the LOG.CSV file containing a startup message and the current time (figure 6.3). After this a delay is built in to prevent data from being gathered while the user is still touching the device or moving around on their chair.

Running

After the setup is complete the device continues with a loop of actions, these actions are repeated four times a second and classify and log the data to a CSV file. It starts of by noting down the Arduino time in millis which is later used to determine the correct delay to the next cycle, following that the data from accelerometer is retrieved and put into a array so that it can be used by the k-NN classifier.

The k-NN classifier then takes the accelerometer data together with the the number of attributes (3) and their names (X, Y, Z), the numbered classifier categories (0 to 4), the amount of rows in the database (100) and the k which is the amount of neighbours it will check. The k-NN classifier works in four steps, first the euclidean distance, which is the amount of distance between two points in a euclidean space, from the new data point to all of the existing data points is calculated and stored in an array. In the second step the distances are ranked in ascending order, after which the third step extracts the amount of neighbours according to the number k. The fourth step counts the frequency of each category and selects that neighbour and the fifth step returns the most prevalent category which is then used to classify the data. This process is also visualized in figure 5.4.

The classified data is then placed in an array which is used to calculate the minute averages of the classification, the time is retrieved and a string is created to which the accelerometer data, the time and date and the category in which the data is classified is added. Every minute the average posture of that minute is calculated and added to the string, after which the string gets written to the CSV file on the SD card (figure 6.3). Finally the Arduino time in millis is noted down again and used to calculate the necessary delay in order to keep a frequency of four Hz, in the event the program takes more than 250 milliseconds to complete the delay is set to 0.

```
-> Initializing SD card...initialization done.
-> Classification of accelerometer data using k-NN
-> Starting new session, 15/32/48, 30/6/2020
-> AccX: 16968,AccY: 536,AccZ: 188, 15/32/53, 30/6/2020, Classified as:, Lying down
```

Figure 6.3: Output of the wearable to the CSV file

6.5 Functional testing

In order to test the functionalities of the whole prototype system including the Breathline wearable (figure 6.4)the author performed various functional tests on himself, these tests were executed to test the various functionalities of both the Lo-Fi and Hi-Fi prototypes. All functionalities of the system were tested extensively, and no problems were found with any of the functionalities of the system.

In order to test the k-NN classifier experiments with various database sizes and k numbers were done, it was found that one data point per static orientation and k = 1 is enough to determine static postures such as sitting upright or slouching forwards or backwards. However, in order to identify movement a larger amount of data points is necessary, for the functional prototype used in the user testing a database size of 100 points will be used with k = 5, which was tested and accurately able to determine movement as well. Because the wearing position of the prototype will vary slightly between users it is necessary that for each user their own set of data points is gathered, these will then be put into the database and with that database the accuracy of the prototype can be tested.

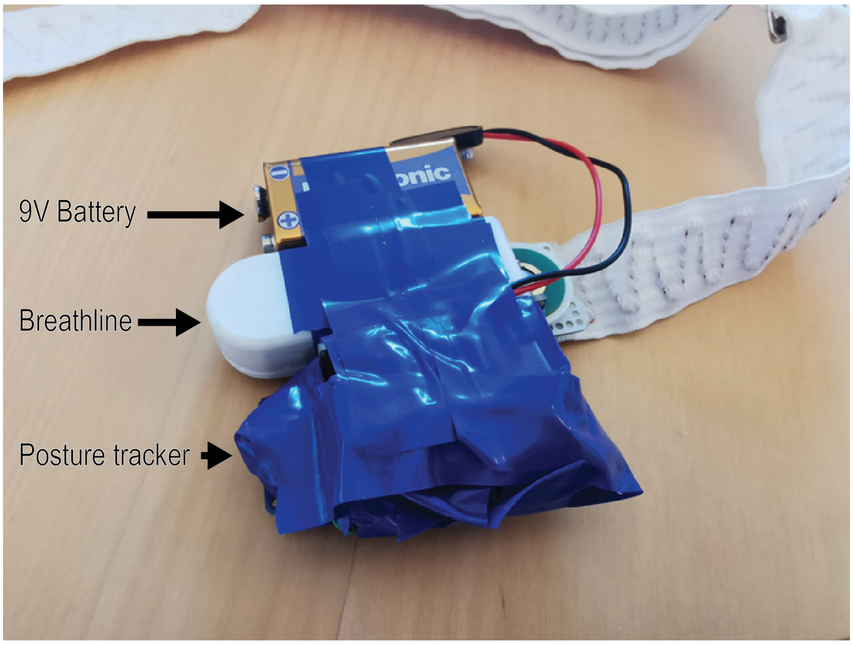

Figure 6.4: Prototype as used in functional and user testing

7 EVALUATION

The goal of this evaluation phase is to answer the fourth sub research question, test the accuracy of the classifier, and check whether all functional and non functional requirements are met. This will be done using three types of evaluation. First a user test will be done to both gather data in order to test the accuracy of the classifier and user feedback on the prototype itself. After which the data gathered during these user test will be used to answer the fourth sub research question: What benefits can be gained from comparing current breathing pattern to the ideal breathing pattern when classifying body position or posture? Finally, the functional and non-functional requirements will be evaluated and a conclusion is given.

7.1 Test procedure

To make sure that information given to the users is consistent a script was created so that every user gets the same amount of information. This is important as it allows for non parametric interview data to be compared between users. The script explains the user testing procedure, setup of the device, the gathering of the personal set of data, the testing procedure of the device, and the questionnaire that they are asked to fill in afterwards.The users are not asked to employ a specific abdominal breathing pattern in order to interfere the least with their planned work during the one hour observation period and in order to get the most consistent breathing data results possible.

Half of the participants will test the the functional prototype without feedback on their posture and the other half will test the functional prototype with feedback on their posture.

First, users are presented with the information brochure and the consent form (appendix F and G). Which they are asked to read and sign. Next, an explanation is given on how the device works, how the personal data set is gathered and how it is worn. The device is then adjusted to the user and put on.

Now the personal data set of the user is gathered, this data set is used for the calibration of the device. The user is asked to sit up straight, forwards, backwards, walk around and lay down for one minute each. Then the user is asked to remove the wearable and the researcher will upload the user specific data set to the wearable. Now the user is requested to wear the device for a period of one hour where the researcher notes down the average posture of the user per one minute for the first 55 minutes, and then for the last five minutes the standard postures are repeated again as a verification test. If posture average for that minute is unclear to the researcher the data point for that minute will not be used.

After this period of one hour the user is asked to remove the device and fill in a questionnaire. There are two questionnaires available depending on if the user tested the device with feedback on posture or without feedback on posture. Half of the participants answers questions about whether they think feedback is necessary and the other half answers questions about the feedback they received from the device. And then the whole group of participants answers a set of general questions.

The questionnaire will consist of several likert scale questions aimed at the functionalities of the device, whether the device is comfortable to wear, their opinion on the calibration process, whether they would use such a device, and if they tested the prototype with feedback whether they found the feedback to be satisfactory.

7.2 Classifier accuracy

The accuracy of the prototype wearable was tested for two different operational modes, the first including classifying movement, and the second only classifying the static positions, excluding movement. Both tests included n = 4 participants, for the second operational mode the optional feedback via vibrations was also included and participants were asked for their opinion on the feedback.

For the accuracy testing of the classifier k = 5 was set in the program, and all participants were tested with their own set of data points that was collected before the test. The results from the initial user testing can be found in table 7.1 and table 7.2. When processing the data, however, some anomalies were found and after investigation into the code an accidental cross-reference of an array was found. In essence, this error lead to a k = 1 nearest neighbour classifier instead of a k = 5 nearest neighbour classifier. This error caused "movement" to be identified many times when the test user was actually sitting still. Because of this the accuracy of the classifier for the first four test users has to be ignored, the test results of the last four test users can still be used as it was concluded in the realisation phase that a single neighbour classifier is enough to accurately detect static postures. After the cross referencing error was fixed participant one was tested again for a period of one hour and the results of that test can be found in table 7.3 as well.

The accuracy of the first four user tests completed with k = 1 neighbour and the movement detection included show an average accuracy of 56.01% (table 7.1) confirming that selecting based on a single nearest neighbour is not a viable solution when trying to detect both static positions as well as movement. The five minute standard posture test performed at the end of the user test shows decent results with an accuracy of 86.67%, participant 1 did not perform the test because of time constraints on the side of the user.

Participant	1	2	3	4	Average	
k =	1	1	1	1		
Include movement detection	Yes	Yes	Yes	Yes		
Data points used	56	54	49	57	Total data points used	216
Data points correctly classified	13	36	41	31	Total data points correctly classified	121
Detection accuracy	23.21%	66.67%	63.27%	71.93%	Average detection accuracy	56.01%
Standard posture test accuracy	-	80%	100%	80%	Average standard posture test accuracy	86.67%

Table 7.1: kNN classifier accuracy for k = 1 and detection categories Upright, Forward, Backward, Movement, Lying down.

The second set of four users was tested without the movement detection enabled, which means that data was only being classified for static positions. As concluded in section 6.5 a single nearest neighbour is enough for static posture detection and an average accuracy of 93.93% was achieved with a minimum accuracy of 84.75% for participant 5 (table 7.2). The four minute standard posture test achieved a very good accuracy of 91.67%. Participant 8 had a partial disconnect of the battery at minute around minute 35, because of this the system failed to record the last 25 minutes of the session including the four minute standard posture test.

Participant	5	6	7	8	Average	
k =	1	1	1	1		
Include movement detection	No	No	No	No		
Data points used	59	59	62	34	Total data points used	214
Data points correctly classified	50	59	60	32	Total data points correctly classified	201
Detection accuracy	84.75%	100%	96.77%	94.12%	Average detection accuracy	93.93%
Standard posture test accuracy	75%	100%	100%	-	Average standard posture test accuracy	91.67%

Table 7.2: kNN classifier accuracy for k = 1 and detection categories Upright, Forward, Backward, Lying down.

To verify that movement detection is possible and reliable the user test was repeated for participant one and eight with k = 5, and an average accuracy of 90.60% (table 7.3) was achieved. For the five minute standard posture test an accuracy of 100% was achieved. These results together with the result from the second set of user tests tell us that k-NN classifying is an accurate method of classifying posture, body position and movement data in real time.

Participant	1	8	Average	
k =	5	5		
Include movement detection	Yes	Yes		
Data points used	58	59	Total data points used	117
Data points correctly classified	53	53	Total data points correctly classified	106
Detection accuracy	91.38%	89.83%	Average detection accuracy	90.60%
Standard posture test accuracy	100%	100%	Average standard posture test accuracy	100%

Table 7.3: kNN classifier accuracy for k = 5 and detection categories Upright, Forward, Backward, Movement, Lying down.

7.3 Breathing data

The breathing data was analyzed together with the posture data by making use of matlab (version r2019b[87]), matlab was used to plot the breathing data and color that data according to the body position at that time, an overview of how a one hour session looks is given in figure 7.1. Sections of this figure will be analysed for each type of body position to see if there are differences in breathing. It is important to note that the breathing patterns analyzed are the participants natural breathing patterns as they were not asked to employ a specific breathing pattern.

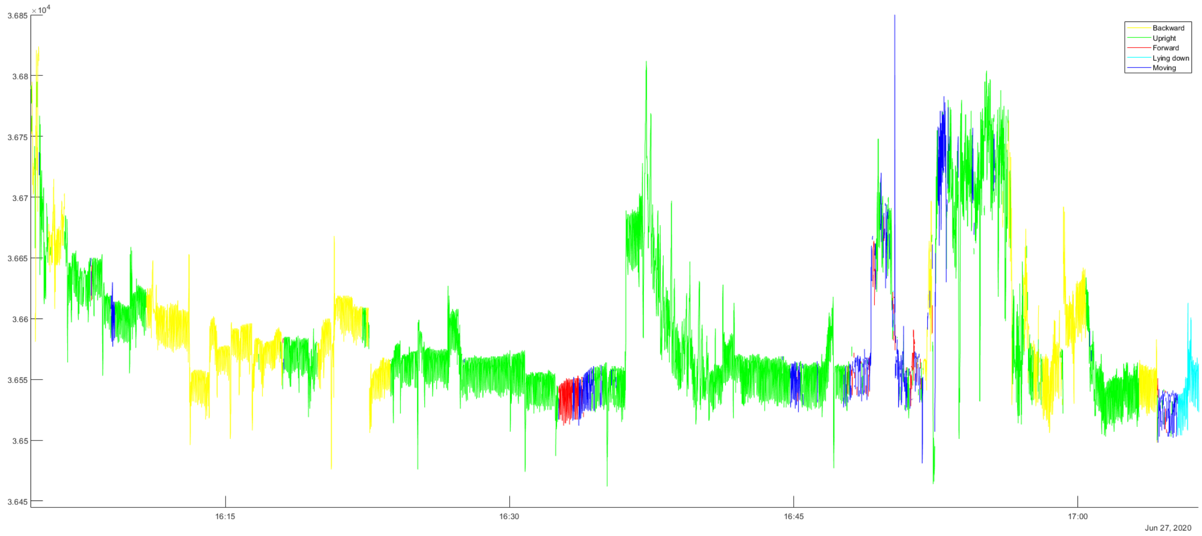

Figure 7.1: One hour breathing graph with green = Upright, yellow = Forward, red = Backward, blue = Moving and cyan = Lying down.

When viewing the data as a whole a few things stand out, first of all there are jumps or drops in the data when the user either adjusts their position or the RIP band, these will need to be filtered out if a full analysis of all data is done. However because only small sections of data will be evaluated on respiration rate(RR) and amplitude(RA) this is not a problem and the baseline drift was not filtered out..

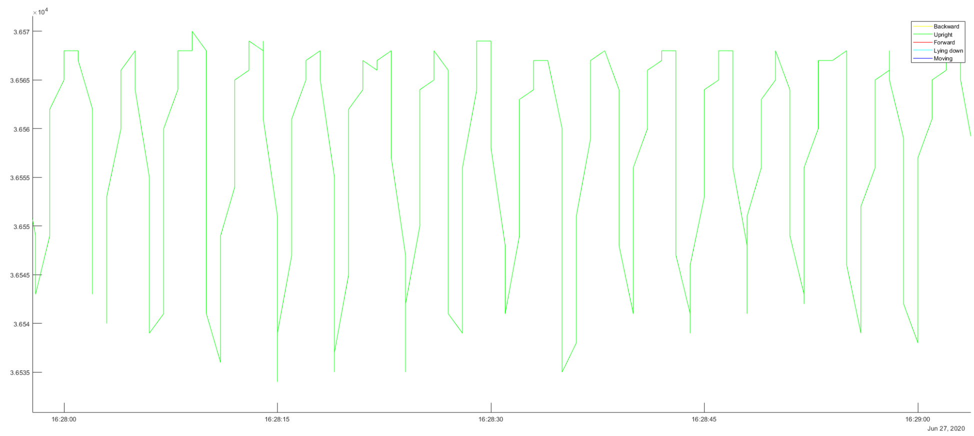

Figure 7.2: One minute sample of Upright breathing.

A one minute sample of Upright breathing is given in figure 7.2, no interpolation is applied to the sample that is why it might not look like a perfect sinus, however this does not effect the respiratory rate and respiratory amplitude analysis. A sample of one minute was taken for each of the body positions for four patients and the respiration rate (breaths per minute) and average respiration amplitude was calculated. These results can be found in table 7.4. The respiration rate and amplitude for movement were excluded as consistent results could not be calculated for most participants, four participants that had all other body positions prevalent in their data were used.

Participant	1	3	4	7
Upright RR	14.3	11.4	15.1	14.9
Backward RR	15.5	10.9	14.7	14.1
Forward RR	14.9	11.2	14.6	15.4
Lying down RR	16.8	14.1	15.1	15.8
Upright RA	29.35	35.27	22.46	31.29
Backward RA	28.72	31.15	26.37	31.42
Forward RA	22.52	36.34	25.43	31.81
Lying down RA	12.43	18.67	14.55	23.76

Table 7.4: Respiratory rate in breaths per minute and respiratory amplitude for different body positions.

This is a small test with only n = 4 samples per participant and because of that these are not statistically significant results. However it can be observed that there seem to be no real differences when comparing upright RR with either backward, forward or lying down RR, and also no real difference between upright RA and backward and forward RA. When comparing upright RA with lying down RA there might be a decrease, however further testing will need to be done in order to get results that are significant. Since it is not possible to find a correlation between body position or posture and breathing frequency or amplitude, the answer to the fourth sub research question at this point is that currently no benefits can be gained from comparing the breathing patterns in order to identify body position or posture.

7.4 User feedback

The participants of the user testing answered a survey on the system. In this survey they were asked general questions about whether they would use a wearable posture correction product, if the system was comfortable to wear, what they thought of the calibration process and the general usability of the device. The group of users that tested the device with feedback were asked if this feedback was well defined, whether it was timed correctly and if they felt like the feedback was correct. The group that was tested without feedback were asked whether they would be more inclined in using the system if it gave feedback. The full results of the survey questions can be found in appendix E. During the user testing comments of the users on the system were also noted down, these comments are also used to further capture the opinion of the users.

Two different questions were asked on the comfort of the device, the results of these questions are that four out of eight participants found the device to be comfortable to wear, two were neutral on the subject and two participants found the device to be uncomfortable. There were also four users who found the device noticeable while wearing, two who were neutral on the subject and only one user who found the device not noticeable while wearing. Two participants mentioned during the testing that they found the device to be quite heavy when walking around specifically. There are definitely possible improvements to be made in the comfort of device, especially in the size and weight of the device.

One question was asked on the calibration process as they perceived it. Four out of eight users found the calibration process to not be cumbersome, three were neutral on the topic and one user found the calibration process to be cumbersome. Five users found the calibration process to be easy, two were neutral on the subject and only one user found the process hard. One user asked why the process of calibration was necessary because she felt it took quite some time.

When it comes to general usability of the device three users mentioned that they found it nice that no other additional actions are required after putting on the device. In the questionnaire four participants found the device easy to use and the other four participants were neutral on the subject.

The feedback of the system was found to be well defined by two participants and two participants were neutral, three participants thought that the feedback was useful in maintaining correct posture and one participant was neutral on the subject. Three out of four participants that did not receive feedback said they would be more inclined to use the system if it would give them feedback. During the user test one user noted that the system gave feedback while she was walking, in the data it was later found that the first half of that minute she was sitting backwards which made the average of that minute backwards. Perhaps a different system of feedback can be imagined to prevent this kind of feedback.

7.5 Evaluation of requirements

The requirements as defined in section 5.4 will be evaluated in this section, this is done making use of the results from the user testing as well as the answers from the survey given to users after testing.

7.5.1 Requirements

Must have

FR1: The system must be able to differentiate between the being upright and laying down body positions.

This requirement was partially met.

The system is able to determine upright and lying down body positions, however the system is not accurately able to determine between standing or sitting upright, that is why this was combined to a general upright position. A second accelerometer could solve this problem.

FR2: The system must be able to detect forward posture, normal posture and backward posture.

This requirement was met.

The system is able to determine normal posture (upright), forward poster and backward posture.

FR3: The system must be able to detect walking.

This requirement was met.

The system is able to determine movement in the form of walking.

FR4: The system must store the data gathered in a CSV file.

This requirement was met.

The system stores both the raw data as well as the classified data in a CSV file on a SD card which can be removed and placed in a computer for further analysis.

FR5: The system must use an Arduino in combination with an accelerometer.

This requirement was met.

The system uses an Arduino Nano Every in combination with a MPU6050 IMU of which only the accelerometer is used.

FR6: The system must be mounted in the abdominal region.

This requirement was met.

The system is mounted in the abdominal region just like the Breathline wearable is allowing for future integration with the Breathline wearable.

FR7: The system must automatically classify the accelerometer data with at least an 80% accuracy.

This requirement was met.

The system classifies the accelerometer data with an average accuracy of 93.93% (table 7.2 when not including movement classification and a 90.60% (table 7.3 accuracy when including movement classification.

FR8: The system must have a manual posture/body position calibration.

This requirement was met.

The system is able to be manually calibrated to a users posture by uploading a data set containing an arbitrary number of accelerometer data points for each body position, posture or movement that is to be classified by the device.

Should have

FR9: The system should have an automatic posture/body position calibration.

This requirement was not met.

The system does not currently have a method to automatically calibrate body position, posture or movement.

FR10: The system should collect data with a frequency of four or eight Hertz.

This requirement was met.

The system is able to collect data with a frequency of four Hertz.

FR11: The system should use a k-Nearest Neighbour classifier for real time classifying of data.

This requirement was met.

The system uses a k-Nearest Neighbour classifier for the analysis of accelerometer data and is able to classify this data in real time.

FR12: The system should allow users to view their raw and classified data.

This requirement was met.

The system allows users to take out the SD card and insert it into a computer in order to view and further analyse the data.

FR13: The system should be able to be run for a whole day without user interference.

This requirement was met.

The system contains a battery and enough storage space in order to run for at least a full day without user interference.

FR14: The system should be able to detect a ten degree change of posture in the upper chest and back.

This requirement was not met.

It was not tested what the accuracy in degrees of the device is for the upper back chest and back.

Could have

FR15: The system could give real time feedback to the users on their sitting posture.

This requirement was met.

The system gives real time feedback to the users in the form of vibrations when their minute average is classified as forward or backward posture.

FR16: The system could have an option to turn the posture and body position sensing functionality off, while keeping the breathing data collection on.

This requirement was partially met.

The system is not yet integrated with the Breathline wearable and therefore its battery can be disconnected in order to turn of the functionality. There is no dedicated switch or other method to turn of this functionality integrated in the device.

NFR1: The system should be comfortable for a user to wear

This requirement was partially met.

It was found during user testing that not all users found the device comfortable to wear. The main complaint was the weight of the system, which needs to be reduced in the future.

7.5.2 General results

In general the requirements are met, there are a few requirements that are partially met such as FR1, the difference between sitting and standing upright cannot be differentiated with a single accelerometer in the abdominal region, this was discussed with the client and accepted as a limitation of the system. Requirement FR9 was not met because of time constraints, it was deemed to be to ambitious and would take too much time away from more important requirements. FR14 was not met because it was not tested how accurate the device is in detecting change in the upper back and chest.FR16 was "partially" met, FR16 was set as an requirement when the system would be integrated with the current Breathline wearable, this is not yet the case because the code and schematics of the Breathline wearable were not available to the author. NFR1 was partially met, some users found the device a bit to big and heavy, this requirement can be expected to be met when the prototype system is integrated with the Breathline wearable.

7.6 Discussion

The main goal of this evaluation was to test the implementation of a k-NN classifier on an Arduino for the real time classification of accelerometer data. Accurate results were gained after the correction of a cross-reference in the code, and k-NN seems to be a valid method of classifying posture using an accelerometer mounted in the abdominal region. The brief analysis of the breathing data was not able to find conclusive evidence for the influence of posture on breathing, because of that the fourth sub research question: What benefits can be gained from comparing current breathing pattern to the ideal breathing pattern when classifying body position or posture? is answered with the following: no benefits in classifying body position or posture can currently be found when comparing ideal breathing pattern to the current breathing pattern.

7.7 Final requirements

A third iteration of the requirements is given with improvements based on the results from the evaluation phase, some of the requirements that are not yet realised are kept as they might be useful to implement in later versions. And one new requirement (FR9) is given based on the results of the user testing.

- Must have
  - FR1: The system must be able to differentiate between the being upright and laying down body positions.
  - FR2: The system must be able to detect forward posture, normal posture and backward posture.
  - FR3: The system must be able to detect walking.
  - FR4: The system must store the data gathered in a CSV file.
  - FR5: The system must use an Arduino in combination with an accelerometer.
  - FR6: The system must be mounted in the abdominal region.
  - FR7: The system must automatically classify the accelerometer data with at least an 80% accuracy.
  - FR8: The system must have a manual posture/body position calibration.
  - FR9: The system must be integrated in the Breathline wearable.
- Should have
  - FR10: The system should have an automatic posture/body position calibration which is activated either through specified motion or a button.
  - FR11: The system should collect data with a frequency of four or eight Hertz.
  - FR12: The system should use a k-Nearest Neighbour classifier for real time classifying of data.
  - FR13: The system should allow users to view their data via a computer.
  - FR14: The system should be able to be run for a whole day without user interference.
  - FR15: The system should give real time feedback to the users on their sitting posture.
  - NFR1: The system should be comfortable for a user to wear.
- Could have
  - FR16: The system could have an option to turn the posture and body position sensing functionality off, while keeping the breathing data collection on.
- Won't have
  - FR17: The system won't have an Artificial Neural Network classifier.
  - FR18: The system won't have a gyroscope.
  - FR19: The system won't be able to detect a ten degree change of posture in the upper chest and back.
  - NFR2: The system won't have a posture score or rating for the user.

8 CONCLUSION & FUTURE WORK

To conclude this graduation project a comprehensive overview is given of the goals and achievements. Following that all sub research questions are answered and discussed once again after which the main research question is answered and discussed. Finally recommendations for future work are given based on the results of this project.

8.1 Conclusion

The first goal at the start of this project was to create a method in which the integrated accelerometer of the Breathline wearable can be used to classify a users posture, body position and movement. The second goal was to integrate this classified data with the Breathline data in order to find possible correlations between posture and breathing. Both of these goals were not fully achieved although several notable achievements have been made.

A wearable system making use of an accelerometer which is mounted in the abdominal region was created that is able to classify a users posture, body position and movement. This system makes use of a k-NN classifier which runs in real time on an Arduino. And the classified data generated by this system can be fused together with the data collected by the Breathline wearable because of the integrated time stamps in both sets of data.

The first part of the literature research was focused on answering the first sub research question: ”What influence do posture, movement and position have on abdominal breathing?”. In literature it was found that when naturally breathing there is a significant difference in breathing ratio between laying down and sitting upright as well as laying down and standing upright. But more importantly it was also found that there still is a significant difference in ratio when employing abdominal breathing, however, the difference is reduced. Lastly, it was found that movement does not influence breathing ratio however a change in posture while moving does. This answer verifies the need to integrate posture sensing capabilities in the Breathline wearable.

The second part of the literature research was focused on answering the question ”What wearable sensors can be used to identify posture, movement and position?”. It was found that IMU’s are seen as the gold standard in human movement tracking, IMU’s are small and have low power requirements, which makes them ideal for integration into wearable products. Accelerometers alone are a good second option but might struggle more with tracking posture while moving. It was also noted that a single sensor setup might struggle with getting correct data while mounted in the abdominal region. The results from the evaluation phase proved that a single accelerometer setup can be used successfully, with however, a few limitations. A single accelerometer mounted in the abdominal region is not able to differentiate between sitting and standing upright, a problem which could be solved by a second accelerometer.

The third sub research question ”What classification methods are available and which methods are usable on a small micro-processor for classifying a person’s posture, movement or position?” was answered in the ideation phase. Multiple methods were explored including linear and non-linear classifiers, it was determined that non-linear methods are preferable for identifying both static postures as well as movements. A k-NN classifier was chosen because it does not require a learning period like ANN classifiers and because of that is easy to calibrate to each individual. k-NN has also been used successfully in literature to classify accelerometer data and offered the most potential when looking at a real time classifier.

During the evaluation phase the fourth and final sub research question was answered through user testing. ”What benefits can be gained from comparing current breathing pattern to the ideal breathing pattern when classifying body position or posture?”. It was concluded that currently no benefits can be gained from comparing the breathing patterns between postures in classifying body position or posture data. This is because no patterns could be found in the differences between upright breathing and forward or backward breathing.

With the sub research questions all answered and summed up it is finally possible to answer the main research question: ”How can a wearable mounted in the abdominal region be used to detect and classify body position, posture and movement data?”. And the answer to this question comes in the form of the wearable product created in this project. The wearable product is powered by an **Arduino** and makes use of an **accelerometer** to gather data. This data is then classified by a **k-Nearest Neighbour classifier** and the result of this classification is saved to a **CSV file**. But most importantly it provides users with the option to **personalise their baseline posture data** and the decision if they want feedback on their posture or not.

8.2 Recommendations for future work

As a final wrap up of this graduation project some recommendations for future work will be given. These recommendations are mainly based on findings of the evaluation and ideation phases and also cover requirements that are not yet met which could boost the project.

First and foremost integration of the posture wearable within the Breathline wearable should be achieved, this will drastically reduce the size of the overall prototype and allow for integration of the posture data with the breathing data. It will increase user comfort and take away some of the weight issues some users complained about.

Automatic calibration of the device could be implemented to make the prototype more user friendly and is a necessity for a final product. The current method of calibration for the device takes about fifteen minutes and requires knowledge of the programming language, which is of course not feasible for a final product.

More exploration on the topic of breathing and posture/body position should be done to find conclusive results on the topic. That research being either conclusive that breathing is something personal which cannot be compared between humans, or finding a correlation somewhere.

Bibliography

[1] C. Lan, S. . Chen, J. . Lai, and A. M. . Wong, "Tai chi chuan in medicine and health promotion," *Evidence-based Complementary and Alternative Medicine*, vol. 2013, 2013. Cited By :40.

[2] Y.-F. Chen, X.-Y. Huang, C.-H. Chien, and J.-F. Cheng, "The effectiveness of diaphragmatic breathing relaxation training for reducing anxiety," *Perspectives in Psychiatric Care*, vol. 53, no. 4, pp. 329–336, 2017.

[3] J. Depiazzi and M. L. Everard, "Dysfunctional breathing and reaching one's physiological limit as causes of exercise-induced dyspnoea," *Breathe*, vol. 12, no. 2, pp. 120–129, 2016.

[4] M. Jones, A. Harvey, L. Marston, and N. O'Connell, "Breathing exercises for dysfunctional breathing/hyperventilation syndrome in adults," *Cochrane Database of Systematic Reviews*, no. 5, 2013.

[5] S.-Z. Wang, S. Li, X.-Y. Xu, G.-P. Lin, L. Shao, Y. Zhao, and T. H. Wang, "Effect of slow abdominal breathing combined with biofeedback on blood pressure and heart rate variability in prehypertension," *The Journal of Alternative and Complementary Medicine*, vol. 16, no. 10, pp. 1039–1045, 2010. PMID: 20954960.

[6] R. Sonpeayung, A. Tantisuwat, T. Klinsophon, and P. Thaveeratitham, "Which body position is the best for chest wall motion in healthy adults? a meta-analysis," *Respiratory Care*, vol. 63, no. 11, pp. 1439–1451, 2018.

[7] H. Kaneko and J. Horie, "Breathing movements of the chest and abdominal wall in healthy subjects," *Respiratory Care*, vol. 57, no. 9, pp. 1442–1451, 2012.

[8] Wikipedia.org, "diaphragmatic breathing." https://encyclopedia.thefreedictionary.com/diaphragmatic+breathing. Accessed: 2020-04-03.

[9] D. P. Ricciardelli, "Diaphragmatic breathing." https://www.drpricciardelli.com/post/diaphragmatic-breathing. Accessed: 2020-04-15.

[10] K. P. Cohen, "Comparison of impedance and inductance ventilation sensors on adults during breathing, motion, and simulated airway obstruction," *IEEE Transactions on Biomedical Engineering*, vol. 44, no. 7, pp. 555–566, 1997. Cited By :79.

[11] T. Rahman, R. Page, C. Page, J.-R. Bonnefoy, T. Cox, and T. H. Shaffer, "pneuRIPTM: A Novel Respiratory Inductance Plethysmography Monitor," *Journal of Medical Devices*, vol. 11, 01 2017. 011010.

[12] Vivonoetics, "Respiratory inductance plethysmography." https://commons.wikimedia.org/w/index.php?curid=12684092. Accessed: 2020-04-17.

[13] Sparkfun, "How an accelerometer works." https://learn.sparkfun.com/tutorials/accelerometer-basics/all. Accessed: 2020-04-17.

[14] McGraw and Hill, “supine. (n.d.) concise dictionary of modern medicine. (2002)..” https://medical-dictionary.thefreedictionary.com/supine. Accessed: 2020-04-10.

[15] Dimensions.guide, “Supine.” https://www.dimensions.guide/element/standing-male-side. Accessed: 2020-04-18.

[16] Dimensions.guide, “Sitting.” https://www.dimensions.guide/element/office-business-people-sitting-side-female. Accessed: 2020-04-18.

[17] Dimensions.guide, “Standing.” https://www.dimensions.guide/element/standing-male-side. Accessed: 2020-04-18.

[18] S. Katz, N. Arish, A. Rokach, and et al., “The effect of body position on pulmonary function: a systematic review,” *BMC Pulm Med 18, 159 (2018)*, 2018.

[19] L. P. D. S. Mendes, D. S. R. Vieira, L. S. Gabriel, G. A. Ribeiro-Samora, A. D. D. Andrade], D. C. Brandao, M. C. Goes, G. A. F. Fregonezi, R. R. Britto, and V. F. Parreira, “Influence of posture, sex, and age on breathing pattern and chest wall motion in healthy subjects,” *Brazilian Journal of Physical Therapy*, 2019.

[20] C. Hagman, C. Janson, A. Malinovschi, H. Hedenstrum, and M. Emtner, “Measuring breathing patterns and respiratory movements with the respiratory movement measuring instrument,” *Clinical Physiology and Functional Imaging*, vol. 36, no. 5, pp. 414–420, 2016.

[21] C. Heyde, H. Leutheuser, B. Eskofier, K. Roecker, and A. Gollhofer, “Respiratory inductance plethysmography-a rationale for validity during exercise,” *Medicine and science in sports and exercise*, vol. 46, pp. 488–495, 02 2014.

[22] E. Bernardi, L. Pratali, G. Mandolesi, M. Spiridonova, G. Roi, and A. Cogo, “Thoraco-abdominal coordination and performance during uphill running at altitude.,” vol. 12, no. 3, p. e0174927, 2017. 0174927.

[23] X. Lai, Q. Liu, W. Wang, and L. . Li, “Research on posture recognition based on kalman filter and quaternion,” in *Proceedings 2012 International Conference on System Science and Engineering, ICSSE 2012*, pp. 211–216, 2012.

[24] Kazusuke Maenaka, “Mems inertial sensors and their applications,” in *2008 5th International Conference on Networked Sensing Systems*, pp. 71–73, June 2008.

[25] S. Z. Homayounfar and T. L. Andrew, “Wearable sensors for monitoring human motion: A review on mechanisms, materials, and challenges,” *SLAS TECHNOLOGY: Translating Life Sciences Innovation*, vol. 25, no. 1, pp. 9–24, 2020. PMID: 31829083.

[26] S. Hanly, “Accelerometers: Taking the guesswork out of accelerometer selection.” https://blog.endaq.com/accelerometer-selection. Accessed: 2020-04-03.

[27] H. J. Luinge and P. H. Veltink, “Measuring orientation of human body segments using miniature gyroscopes and accelerometers,” *Medical and Biological Engineering and Computing*, vol. 43, no. 2, pp. 273–282, 2005. Cited By :431.

[28] K. Thaler-Kall, F. Tusker, J. Hermsdörfer, L. Gorzelniak, and A. Horsch, *Where to wear accelerometers to measure physical activity in people?*, vol. 192 of *Studies in Health Technology and Informatics*. 2013.

[29] H. Maki, Y. Yonezawa, H. Ogawa, A. W. Hahn, and W. M. Caldwell, “A wearable data acquisition system employing one tri-axis accelerometer,” *Biomedical sciences instrumentation*, vol. 45, pp. 424–429, 2009.

[30] V. C. F. Chen and S. . Chen, “Establishing the waist as the better location for attaching a single accelerometer to estimate center of pressure trajectories,” *Clinical Biomechanics*, vol. 60, pp. 30–38, 2018. Cited By :1.

[31] D. Arvidsson, J. Fridolfsson, and M. Börjesson, “Measurement of physical activity in clinical practice using accelerometers,” *Journal of internal medicine*, vol. 286, no. 2, pp. 137–153, 2019. Cited By :2.

[32] I. C. Gyllensten and A. G. Bonomi, “Identifying types of physical activity with a single accelerometer: Evaluating laboratory-trained algorithms in daily life,” *IEEE Transactions on Biomedical Engineering*, vol. 58, no. 9, pp. 2656–2663, 2011. Cited By :98.

[33] E. Fortune, V. A. Lugade, and K. R. Kaufman, “Posture and movement classification: The comparison of tri-axial accelerometer numbers and anatomical placement,” *Journal of Biomechanical Engineering*, vol. 136, no. 5, 2014. Cited By :15.

[34] N. Jason, A. V. Sahakian, and S. Swiryn, “Accelerometer-based body-position sensing for ambulatory electrocardiographic monitoring,” *Biomedical Instrumentation and Technology*, vol. 37, no. 5, pp. 338–346, 2003. Cited By :14.

[35] B. Kemp, A. J. Janssen, and B. [van der Kamp], “Body position can be monitored in 3d using miniature accelerometers and earth-magnetic field sensors,” *Electroencephalography and Clinical Neurophysiology/Electromyography and Motor Control*, vol. 109, no. 6, pp. 484 – 488, 1998.

[36] X. Yao, G. Sun, W. . Lin, and W. . Chou, “The design of a real-time accelerometer-based sleeping position monitoring system and its application on obstructive sleep apnea syndrome,” in *2012 International Conference on Systems and Informatics, ICSAI 2012*, pp. 1061–1066, 2012. Cited By :2.

[37] M. Trächtler, D. Hodgins, L. Kenney, M. Dienger, T. Link, and Y. Manoli, “Multi-axis inertial measurement units measuring human posture and motion,” in *IFMBE Proceedings*, vol. 13, pp. 184–188, 2007.

[38] A. Efimovskaya, *Multi-Axis Solutions for MEMS Inertial Sensors.* PhD thesis, UC Irvine, 2017.

[39] H. Zhang, X. Ke, and R. Jiao, “Experimental research on feedback kalman model of mems gyroscope,” in *2007 8th International Conference on Electronic Measurement and Instruments, ICEMI*, pp. 1253–1256, 2007. Cited By :5.

[40] K. Ben Mansour, N. Rezzoug, and P. Gorce, “Comparison between several locations of gyroscope for gait events detection,” *Computer methods in biomechanics and biomedical engineering*, vol. 18, pp. 1996–1997, 2015. Cited By :2.

[41] F. Höflinger, J. Müller, M. Törk, L. M. Reindl, and W. Burgard, “A wireless micro inertial measurement unit (imu),” in *2012 IEEE International Instrumentation and Measurement Technology Conference Proceedings*, pp. 2578–2583, May 2012.

[42] M. Perlmutter and L. Robin, “High-performance, low cost inertial mems: A market in motion!,” in *Proceedings of the 2012 IEEE/ION Position, Location and Navigation Symposium*, pp. 225–229, 2012.

[43] A. Petropoulos, D. Sikeridis, and T. Antonakopoulos, “Spomo: Imu-based real-time sitting posture monitoring,” in *IEEE International Conference on Consumer Electronics - Berlin, ICCE-Berlin*, vol. 2017-September, pp. 5–9, 2017. Cited By :2.

[44] J. E. Estrada and L. A. Vea, “Real-time human sitting posture detection using mobile devices,” in *Proceedings - 2016 IEEE Region 10 Symposium, TENSYMP 2016*, pp. 140–144, 2016. Cited By :7.

[45] S. Shin and W. Yoo, “Inertial measurement unit-based evaluation of global and regional lumbar spine and pelvis alignment in standing individuals with a flat lumbar posture,” *Journal of manipulative and physiological therapeutics*, vol. 42, no. 8, pp. 594–600, 2019.

[46] S. Tanaka, K. Motoi, M. Nogawa, and K. Yamakoshi, “A new portable device for ambulatory monitoring of human posture and walking velocity using miniature accelerometers and gyroscope,” in *Annual International Conference of the IEEE Engineering in Medicine and Biology - Proceedings*, vol. 26 III, pp. 2283–2286, 2004. Cited By :29.

[47] M. Airaksinen, O. Räsänen, E. Ilén, T. Häyrinen, A. Kivi, V. Marchi, A. Gallen, S. Blom, A. Varhe, N. Kaartinen, L. Haataja, and S. Vanhatalo, “Automatic posture and movement tracking of infants with wearable movement sensors,” *Scientific Reports*, vol. 10, no. 1, 2020.

[48] A. Vial, D. Stirling, M. Ros, P. J. Vial, and M. Field, “Modelling dynamic body motion using imus positioned on the torso,” in *2017, 11th International Conference on Signal Processing and Communication Systems, ICSPCS 2017 - Proceedings*, vol. 2018-January, pp. 1–6, 2018. Cited By :1.

[49] B. Najafi, K. Aminian, A. Paraschiv-Ionescu, F. Loew, C. J. Bula, and P. Robert, “Ambulatory system for human motion analysis using a kinematic sensor: monitoring of daily physical activity in the elderly,” *IEEE Transactions on Biomedical Engineering*, vol. 50, no. 6, pp. 711–723, 2003.

[50] Z. Zheng, X. Li, and S. Tateno, “Development of postures estimation system for walking training,” in *2018 International Automatic Control Conference, CACS 2018*, 2019.

[51] S. Yang and Q. Li, “Imu-based ambulatory walking speed estimation in constrained treadmill and overground walking,” *Computer methods in biomechanics and biomedical engineering*, vol. 15, no. 3, pp. 313–322, 2012.

[52] L. Simpson, M. M. Maharaj, and R. J. Mobbs, “The role of wearables in spinal posture analysis: A systematic review,” *BMC Musculoskeletal Disorders*, vol. 20, no. 1, 2019.

[53] Prana, “Prana Tech.” http://prana.co/. Accessed: 2020-04-09.

[54] Persidsky and Ahlund, “System and method to monitor, guide, and evaluate breathing, utilizing posture and diaphragm sensor signals,” 2015. US patent: US20150342518.

[55] BreathBalanz, “BreathBalanz.” https://breathbalanz.com/en/home-en/. Accessed: 2020-04-09.

[56] Spire, “Spire Health Tag.” https://spirehealth.com/pages/research-platform. Accessed: 2020-04-09.

[57] Lumo, “Lumo Lift.” https://www.lumobodytech.com/lumo-lift/. Accessed: 2020-04-09.

[58] Adafruit, “Open it up! Lumo Lift teardown.” https://learn.adafruit.com/lumo-lift-teardown. Accessed: 2020-04-09.

[59] Upright, “Upright GO 1 & 2.” https://www.uprightpose.com/. Accessed: 2020-04-09.

[60] Upright, “Upright FAQ.” https://www.uprightpose.com/blog/questions-asked-and-answered-about-upright-posture-wearables/. Accessed: 2020-04-09.

[61] Alex+, “Posture 360.” https://alexposture.com/. Accessed: 2020-04-09.

[62] N. K. M. Yoong, J. Perring, and R. J. Mobbs, “Commercial postural devices: A review,” *Sensors*, vol. 19, no. 23, 2019.

[63] Opter, “Opter Pose.” https://opterlife.com/. Accessed: 2020-04-09.

[64] Beurer, “8sense.” https://8sense.com/en/. Accessed: 2020-04-09.

[65] TruPosture, “Adela Health.” https://www.truposture.com/. Accessed: 2020-04-09.

[66] A. Mader and W. Eggink, “A design process for creative technology,” in *Proceedings of the 16th International conference on Engineering and Product Design, E&PDE 2014*, E&PDE, pp. 568–573, The Design Society, 9 2014.

[67] S. Bailur, “Using stakeholder theory to analyze telecenter projects,” *Information Technologies and International Development*, vol. 3, pp. 61–80, 03 2007.

[68] N. Bonnardel and J. Didier, “Brainstorming variants to favor creative design,” *Applied Ergonomics*, vol. 83, p. 102987, 2020.

[69] Cleverism, “Brainstorming techniques for idea generation.” https://www.cleverism.com/brainstorming-techniques-for-idea-generation. Accessed: 2020-04-20.

[70] A. Anderson, “Pact analysis and prototype design for an interactive system.” Coventry University, 2011.

[71] N. Bevan, J. Carter, J. Earthy, T. Geis, and S. Harker, “What are user requirements? developing an iso standard,” 06 2018.

[72] Agilebusiness.org, “Moscow prioritisation.” https://www.agilebusiness.org/page/ProjectFramework_10_MoSCoWPrioritisation. Accessed: 2020-05-07.

[73] W. Heuten, H. Hermens, J.-D. Hoffmann, M. Tabak, J. Timmermann, D. Willemsen, A. Workowski, and J. Technau, *Coaching of Body Awareness Through an App-Guide: The HealthNavigator*, pp. 191–203. Cham: Springer International Publishing, 2017.

[74] Covington, “Are wearables medical devices requiring a ce-mark in the eu?.” https://www.covingtondigitalhealth.com/2019/01/are-wearables-medical-devices-requiring-a-ce-mark-in-the-eu/. Accessed: 2020-05-11.

[75] TÜV, “Tüv.” https://www.tuv.nl/nl/home/. Accessed: 2020-06-29.

[76] X. Huang, F. Wang, J. Zhang, Z. Hu, and J. Jin, “A posture recognition method based on indoor positioning technology,” *Sensors*, vol. 19, p. 1464, 03 2019.

[77] C. Mattmann, O. Amft, H. Harms, G. Troster, and F. Clemens, “Recognizing upper body postures using textile strain sensors,” in *2007 11th IEEE International Symposium on Wearable Computers*, pp. 29–36, 2007.

[78] H. Z. Tan, L. A. Slivovsky, and A. Pentland, “A sensing chair using pressure distribution sensors,” *IEEE/ASME Transactions on Mechatronics*, vol. 6, no. 3, pp. 261–268, 2001.

[79] statisctics4u.com, “Classification vs calibration.” http://www.statistics4u.com/fundstat_eng/cc_classif_calib.html. Accessed: 2020-06-03.

[80] S. Kaghyan and H. Sarukhanyan, “Activity recognition using k-nearest neighbor algorithm on smartphone with triaxial accelerometer,” *International Journal of Informatics Models and Analysis (IJIMA), ITHEA International Scientific Society*, vol. 1, pp. 146–156, 01 2012.

[81] Scikit-learn, “Knn classification using scikit-learn.” https://www.datacamp.com/community/tutorials/k-nearest-neighbor-classification-scikit-learn. Accessed: 2020-07-01.

[82] Arduino, “Arduino nano every.” https://store.arduino.cc/arduino-nano-every. Accessed: 2020-06-30.

[83] Invensense, “Mpu6050.” https://invensense.tdk.com/products/motion-tracking/6-axis/mpu-6050/. Accessed: 2020-06-30.

[84] Tinytronics, “Micro sd card adapter.” https://www.tinytronics.nl/shop/nl/diversen/opslag/microsd-kaart-adapter-module-3.3v-5v-met-level-shifter. Accessed: 2020-06-30.

[85] Opencircuit, “Real time clock.” https://opencircuit.nl/Product/Real-time-clock-module-DS3231. Accessed: 2020-06-30.

[86] Fritzing.org, “Fritzing.” https://fritzing.org/home/. Accessed: 2020-06-14.

[87] Mathworks, “Mathworks matlab.” https://nl.mathworks.com/products/matlab.html. Accessed: 2020-07-03.

A APPENDIX

Listing A.1: Lo-Fi prototype 1: main code

```
// MPU-6050 Short Example Sketch
// By Arduino User JohnChi
// August 17, 2014
// Public Domain
// Adapted by Martijn Poot to only read accelerometer data.

#include<Wire.h>
const int MPU_addr=0x68;  // I2C address of the MPU-6050
int16_t AcX,AcY,AcZ;
void setup(){
  Wire.begin();
  Wire.beginTransmission(MPU_addr);
  Wire.write(0x6B);  // PWR_MGMT_1 register
  Wire.write(0);     // set to zero (wakes up the MPU-6050)
  Wire.endTransmission(true);
  Serial.begin(9600);
}
void loop(){
  Wire.beginTransmission(MPU_addr);
  Wire.write(0x3B);  // starting with register 0x3B (ACCEL_XOUT_H)
  Wire.endTransmission(false);
  Wire.requestFrom(MPU_addr,14,true);  // request a total of 14 registers
  AcX=Wire.read()<<8|Wire.read();  // 0x3B (ACCEL_XOUT_H) & 0x3C (ACCEL_XOUT_L)
  AcY=Wire.read()<<8|Wire.read();  // 0x3D (ACCEL_YOUT_H) & 0x3E (ACCEL_YOUT_L)
  AcZ=Wire.read()<<8|Wire.read();  // 0x3F (ACCEL_ZOUT_H) & 0x40 (ACCEL_ZOUT_L)
  Serial.print("AcX = "); Serial.print(AcX);
  Serial.print(" | AcY = "); Serial.print(AcY);
  Serial.print(" | AcZ = "); Serial.println(AcZ);
  delay(100);
}
```

B APPENDIX

Listing B.1: Lo-Fi prototype 2: main code

```
// k-Nearest Neighbour classification for accelerometer data //
// Author: Martijn Poot //
// //

#include "dataset.h"
#include "kNNFunctions.h"
float AccData[NoAttr];
long beforeCalc = 0, afterCalc = 0;

//Printing stuffs

void printing(float AccData[], int Classified){

  //Serial.println("-----------------------------------------------");
  //Serial.print(AttributeNames[0]); Serial.print(": "); Serial.print(AccData[0],3); Serial.println(" cms");
  //Serial.print(AttributeNames[1]); Serial.print(": "); Serial.print(AccData[1],3); Serial.println(" cms");
  //Serial.print(AttributeNames[2]); Serial.print(": "); Serial.print(AccData[2],3); Serial.println(" cms");
  Serial.print(" >> Classified as: "); Serial.println(Catagories[Classified]);
}


// Setup the arduino

void setup() {
  Serial.begin(9600);
  delay(1000);
  Serial.println("Classification of accelerometer data using k-NN");
  delay(500);
}


// Main program

void loop() {
```

```

  // place all accelerometer data in a array
  AccData[0] = AcX; AccData[1] = AcY; AccData[2] = AcZ;

  // tic (this is for speed calculation)
  beforeCalc = millis();


  // use the classification algorithm to classify the incoming data
  int Classified = kNNClassifier((float*)Attributes,
      CatagoriesNumbered, AccData, 1, NoRow, NoAttr);

  // tac (this is for speed calculation)
  afterCalc = millis();

  //execute a void that just prints stuff
  printing(AccData, Classified);
  //how fast was the classifier?
  Serial.print("Calculation time: "); Serial.print(afterCalc-
      beforeCalc); Serial.println(" mS");

  delay(1000);
}
```

Listing B.2: Lo-Fi prototype 2: database

```
//Database containing the reference dataset, this data is used as the
    "neighbours"

const int NoRow = 4; //  Number of rows in the database
const int NoAttr = 3; //  Number of attributes per row

// Classes alphanumeric database
char* Catagories[] = {"Normal", "Forward", "Backward", "Lying down"};

//  Attributes: training set
char* AttributeNames[]={"X", "Y", "Z"};

//Data points // add in accelerometer data here // example: {1060,
    264, 18000},
float Attributes[NoRow][NoAttr] = {

};

//catagory for each data point, done in numbers for memory size
    reduction // example: 0,
int CatagoriesNumbered[NoRow] = {

};
```

Listing B.3: Lo-Fi prototype 2: k-NN classifier

```
#include <HardwareSerial.h> //   Debug only

//functions for the computation algorithm KNN

//#include "dataset.h"
#include "kNNFunctions.h"

#include <math.h>

//Main part, execute all calculations and return most frequent occuring neighbour

int kNNClassifier(float *Data, int CatagoriesNumbered[], float AccData[], int k, int NoRow, int NoAttr){
  float Extra2[NoRow], FirstNeigbour[k];
  int Catagories[NoRow];
  int kNumber[k];

  TotalEuclideanDistance(AccData, Extra2, (float*)Data, NoRow, NoAttr);
  Ordering(Extra2, Catagories, CatagoriesNumbered, NoRow);
  ExtractFirstNeigbour(Extra2, FirstNeigbour, Catagories, kNumber, k);

  // Calculate Fashion
  int cont = 0, cont2 = 0, pos = 0, num = 0, i = 0;
  int Frequency[k], mayor = 0, posmayor = 0, aux[k];

  // Initialize the frequency counter
  for(i=0; i<k; i++){
    Frequency[k] = 0;
  }

  //Check repetitions of each number
  for(cont=0; cont<k; cont++){
    num = Catagories[cont];
    pos = cont;

    for(cont2 = 0; cont2<k; cont2++){
      if(Catagories[cont2] == num){
        aux[pos]++;
      }
    }
  }

  mayor =aux[0];
  posmayor = 0;

  for(cont=0; cont<k; cont++){
    if(aux[cont] > mayor){
      posmayor = cont;
```

```
      mayor = aux[cont];
    }
  }

  return Catagories[posmayor];
}
// //////////////////////////////////////////////////////////////////

//calculating Euclidean distance between two points

float EuclideanDistance(float pt1[], float pt2[], int NoAttr){
  int i;
  float sum = 0;
  for(i=0; i<NoAttr; i++){
    sum = pow(pt1[i] - pt2[i], 2) + sum;
  }
  return sqrt(sum);
}
// //////////////////////////////////////////////////////////////////

//calculation of Euclidean distance between a point and the database

void TotalEuclideanDistance(float pt1[], float aux[], float *
    Attributes, int NoRow, int NoAttr){
  int i = 0, j = 0;
  float pt2[NoAttr];

  for(i = 0; i<NoRow; i++){
    for(j=0; j<NoAttr; j++){
      pt2[j] = Attributes[i*NoAttr+j];
    }

    aux[i] = EuclideanDistance(pt1, pt2, NoAttr);
  }
}
// //////////////////////////////////////////////////////////////////

//Sorting distances Ascending

void Ordering(float Data[], int Catagories[], int clasesNo[], int
    NoRow){
  int i =1, j = 1, f = 1, temp[2];

  //  Create a copy of the original classes
  for(i=0; i<NoRow; i++){
```

```
    Catagories[i] = clasesNo[i];
  }

  //Ordering
  for(i=1; (i<=NoRow)&&f; i++){
    f = 0;
    for(j=0; j<(NoRow-1); j++){
      if(Data[j+1] < Data[j]){ // descending order >, ascending order
          <
        temp[0] = Data[j];     temp[1] = Catagories[j];
        Data[j] = Data[j+1];  Catagories[j] = Catagories[j+1];
        Data[j+1] = temp[0];   Catagories[j+1] = temp[1];
        f = 1;
      }
    }
  }
}
//
    ///////////////////////////////////////////////////////////////////////////

//Extracting the first N

void ExtractFirstNeigbour(float Data[], float FirstNeigbour[], int
    Catagories[], int kNumber[], int k){
  for(int i=0; i<k; i++){
    FirstNeigbour[i] = Data[i];
    kNumber[i] = Catagories[i];
  }
}
//
    ///////////////////////////////////////////////////////////////////////////

//Calculate the most common type: Fashion

int NeighbourFrequency(int Catagories[], int k){

  int cont = 0, cont2 = 0, pos = 0, num = 0, i = 0;
  int Frequency[k], mayor = 0, posmayor = 0, aux[k];

  //  Initialize the frequency counter
  for(i=0; i<k; i++){
    Frequency[k] = 0;
  }

  // Check repetitions of each number
  for(cont=0; cont<k; cont++){
    num = Catagories[cont];
    pos = cont;

```

```
    for(cont2 = 0; cont2<k; cont2++){
      if(Catagories[cont2] == num){
        aux[pos]++;
      }
    }
  }

  mayor =aux[0];
  posmayor = 0;

  for(cont=0; cont<k; cont++){
    if(aux[cont] > mayor){
      posmayor = cont;
      mayor = aux[cont];
    }
  }

  return Catagories[posmayor];
}
//////////////////////////////////////////////////////////////////
```

Listing B.4: Lo-Fi prototype 2: k-NN classifier.h

```
#ifndef kNNFunctions_H
#define kNNFunctions_H

#include <Arduino.h>
//functions for the computation algorithm k-NN


//calculating Euclidean distance between two points

float EuclideanDistance(float pt1[], float pt2[], int NoAttr);
//////////////////////////////////////////////////////////////////

//calculation of Euclidean distance between a point and the database

void TotalEuclideanDistance(float pt1[], float aux[], float *
    Attributes, int NoRow, int NoAttr);
//////////////////////////////////////////////////////////////////

//Sorting distances Ascending preserving class

void Ordering(float Data[], int clases[], int clasesNo[], int NoRow);
//////////////////////////////////////////////////////////////////
```

```
//Extracting the first N

void ExtractFirstNeigbour(float Data[], float FirstNeigbour[], int clases[],
                          int kClases[], int k);
// ////////////////////////////////////////////////////////////////////////////

//Calculate the most common type: Fashion

int NeighbourFrequency(int clases[], int k);
// ////////////////////////////////////////////////////////////////////////////

//Maps a class integer value to a string

char* IntegerToString(int claseint);
// ////////////////////////////////////////////////////////////////////////////

//Calculate the class of a set of attributes using KNN

int kNNClassifier(float *Data, int clasesNum[], float AccData[], int k, int NoRow, int NoAttr);
// ////////////////////////////////////////////////////////////////////////////

#endif
```

C APPENDIX

Listing C.1: Lo-Fi prototype 3: main code

```

/*
 SD card read/write

  created   Nov 2010
  by David A. Mellis
  modified 9 Apr 2012
  by Tom Igoe
  Modified 3 June 2020
  by Martijn Poot to use CSV instead
  This example code is in the public domain.

*/

#include <SPI.h>
#include <SD.h>

File myFile;

void setup() {
  // Open serial communications and wait for port to open:
  Serial.begin(9600);
  while (!Serial) {
    ; // wait for serial port to connect. Needed for native USB port
        only
  }


  Serial.print("Initializing SD card...");

  if (!SD.begin(4)) {
    Serial.println("initialization failed!");
    while (1);
  }
  Serial.println("initialization done.");

  // open the file. note that only one file can be open at a time,
  // so you have to close this one before opening another.
  myFile = SD.open("test.csv", FILE_WRITE);

```

```
  // if the file opened okay, write to it:
  if (myFile) {
    Serial.print("Writing to test.csv...");
    myFile.println("testing 1, 2, 3.");
    // close the file:
    myFile.close();
    Serial.println("done.");
  } else {
    // if the file didn't open, print an error:
    Serial.println("error opening test.csv");
  }

  // re-open the file for reading:
  myFile = SD.open("test.csv");
  if (myFile) {
    Serial.println("test.csv:");

    // read from the file until there's nothing else in it:
    while (myFile.available()) {
      Serial.write(myFile.read());
    }
    // close the file:
    myFile.close();
  } else {
    // if the file didn't open, print an error:
    Serial.println("error opening test.csv");
  }
}

void loop() {
  // nothing happens after setup
}
```

D APPENDIX

Listing D.1: Hi-Fi prototype code: Main code

```
// k-Nearest Neighbour classification for accelerometer data
// Author: Martijn Poot, June 2020.
// With inspiration from sketches by:
// JohnChi (acclerometer data retrieval)
// Arduino examples library (SD card writing)
// Petre Rodan (Real time clock library)
//

#include "dataset.h" //dataset for Neighbours
#include "kNNFunctions.h" //k-NN calculations
#include "ds3231.h" //Real time clock library
#include <SPI.h> //spi library
#include <SD.h> //SD library
#include<Wire.h> //Wire library

const int MPU_addr=0x69;  // I2C address of the MPU-6050
int16_t AcX,AcY,AcZ; // for the IMU, only using accelerometer
float AccData[NoAttr]; // array for accelerometer output
long beforeCalc = 0, afterCalc = 0; //Speed calculation for the
    script
struct ts t; //for the real time clock
uint8_t Timer = 1;        // the sleep interval in minutes between 2
    consecutive alarms
int ClassifiedArray[500], NextElement = 0; // array and variable for
    minute average calculation

// Setup the arduino
void setup() {
  Wire.begin();
  Wire.beginTransmission(MPU_addr); //start transmission with MPU6050
  Wire.write(0x6B);  // PWR_MGMT_1 register
  Wire.write(0);     // set to zero (wakes up the MPU-6050)
  Wire.endTransmission(true);
  Serial.begin(9600);
  delay(1000); //wait a second for everything to start up (improves
      reliability)

  //catch it if there is no SD card
  Serial.print("Initializing SD card...");
  if (!SD.begin(10)) {
```

```
    Serial.println("initialization failed!");
  while (1);
  }
  Serial.println("initialization done.");
  Serial.println("Classification of accelerometer data using k-NN");

  //start up real time clock and set a timer for 1 minute
  DS3231_init(DS3231_CONTROL_INTCN);
  DS3231_clear_a2f();
  setNextTimer();

  // get time and write to log as a starting message
  DS3231_get(&t);
  String dataString = "Starting new session, ";
  dataString += String(t.hour);
  dataString += "/";
  dataString += String(t.min);
  dataString += "/";
  dataString += String(t.sec);
  dataString += ", ";
  dataString += String(t.mday);
  dataString += "/";
  dataString += String(t.mon);
  dataString += "/";
  dataString += String(t.year);

  File dataFile = SD.open("log.csv", FILE_WRITE);

  // if the file is available, write to it:
  if (dataFile) {
    dataFile.println(dataString);
    //close file again, really important otherwise file corrupts
    dataFile.close();
    // print to the serial port too:
    Serial.println(dataString);
  }
  // if the file isn't open, pop up an error:
  else {
    Serial.println("error opening datalog.txt");
  }
  pinMode( 6 , OUTPUT);  // Must be a PWM pin this is for vibration
     feedback

  //adjust this to whatever time people need to get into position
  delay(5000);
}

// Main program

void loop() {
  // tic (this is for speed calculation)
```

```
beforeCalc = millis();

//start retrieving data from accelerometer
Wire.beginTransmission(MPU_addr);
Wire.write(0x3B);  // starting with register 0x3B (ACCEL_XOUT_H)
Wire.endTransmission(false);
Wire.requestFrom(MPU_addr,6,true);  // request a total of 6 registers
AcX=Wire.read()<<8|Wire.read();  // 0x3B (ACCEL_XOUT_H) & 0x3C (ACCEL_XOUT_L)
AcY=Wire.read()<<8|Wire.read();  // 0x3D (ACCEL_YOUT_H) & 0x3E (ACCEL_YOUT_L)
AcZ=Wire.read()<<8|Wire.read();  // 0x3F (ACCEL_ZOUT_H) & 0x40 (ACCEL_ZOUT_L)
Wire.endTransmission();

// place all accelerometer data in an array
AccData[0] = AcX; AccData[1] = AcY; AccData[2] = AcZ;

// use the classification algorithm to classify the incoming data
int Classified = kNNClassifier((float*)Attributes, CatagoriesNumbered, AccData, 11, NoRow, NoAttr);

// place all postures in array and cycle through array
ClassifiedArray[NextElement] = Classified;
NextElement += 1;

DS3231_get(&t); // Get time from RTC

// make a string for assembling the data to log:
String dataString = "";
dataString += "{";     //"AccX: ";
dataString += String(AcX);
dataString += ", ";    //",AccY: ";
dataString += String(AcY);
dataString += ", ";    //",AccZ: ";
dataString += String(AcZ);
dataString += "}, //" ;     //", ";
dataString += String(t.hour);
dataString += "/";
dataString += String(t.min);
dataString += "/";
dataString += String(t.sec);
dataString += ", ";
dataString += String(t.mday);
dataString += "/";
dataString += String(t.mon);
dataString += "/";
dataString += String(t.year);
dataString += ", Classified as:, ";
dataString += Catagories[Classified]; //add posture classification
```

```
  //timer that is true every minute
  if (DS3231_triggered_a2()) {
    // calculate most frequent posture in that minute and add to dataString
    dataString += ", 1 minute passed: ";
    int MinuteAverage = NeighbourFrequency(ClassifiedArray, NextElement);

    dataString += Catagories[MinuteAverage];
    //Notify user of bad posture via vibration
    if (MinuteAverage == 1 || MinuteAverage == 2){
      analogWrite( 6 , 153 );  // 60% duty cycle
      delay(500);              // play for 0.5s
      analogWrite( 6 , 0 );    // 0% duty cycle (off)
      }
    //reset array again
    memset(ClassifiedArray, 0, sizeof(ClassifiedArray));
    NextElement = 0;
    //set timer for next minute
    setNextTimer();
    // clear a2 alarm flag
    DS3231_clear_a2f();
    }

  // open file
  File dataFile = SD.open("log.csv", FILE_WRITE);

  // if the file is available, write to it:
  if (dataFile) {
    dataFile.println(dataString);
    //close file again, really important otherwise file corrupts
    dataFile.close();
    // print to the serial port too:
    Serial.println(dataString);
  }
  // if the file isn't open, pop up an error:
  else {
    Serial.println("error opening datalog.txt");
  }

   // tac (this is for speed calculation)
  afterCalc = millis();
  int delayCalc = afterCalc-beforeCalc;
  //how fast was the program?
  Serial.print("Calculation time: "); Serial.print(delayCalc); Serial.println(" mS");
  int delayTime = 250-delayCalc;
  if (delayTime < 0){delayTime = 0;}
  delay(delayTime);
}
```

```

// timer code from Petre Rodan (Real time clock library)
void setNextTimer(void)
{
    struct ts t;
    unsigned char wakeup_min;

    DS3231_get(&t);

    // calculate the minute when the next alarm will be triggered
    wakeup_min = (t.min / Timer + 1) * Timer;
    if (wakeup_min > 59) {
        wakeup_min -= 60;
    }

    // flags define what calendar component to be checked against the current time in order
    // to trigger the alarm
    // A2M2 (minutes) (0 to enable, 1 to disable)
    // A2M3 (hour)    (0 to enable, 1 to disable)
    // A2M4 (day)     (0 to enable, 1 to disable)
    // DY/DT          (dayofweek == 1/dayofmonth == 0)
    uint8_t flags[4] = { 0, 1, 1, 1 };

    // set Alarm2. only the minute is set since we ignore the hour and day component
    DS3231_set_a2(wakeup_min, 0, 0, flags);

    // activate Alarm2
    DS3231_set_creg(DS3231_CONTROL_INTCN | DS3231_CONTROL_A2IE);
}
```

Listing D.2: Hi-Fi prototype code: k-NN classifier library

```
#include <HardwareSerial.h> //  Debug only

//functions for the computation algorithm KNN

//#include "dataset.h"
#include "kNNFunctions.h"

#include <math.h>

//Main part, execute all calculations and return most frequent occuring neighbour

int kNNClassifier(float *Data, int CatagoriesNumbered[], float AccData[], int k, int NoRow, int NoAttr){
  float Extra2[NoRow], FirstNeigbour[k];
  int Catagories[NoRow];
  int kNumber[k];

```

```
  //first calculate all distances
  TotalEuclideanDistance(AccData, Extra2, (float*)Data, NoRow, NoAttr);
  //then order them ascending
  Ordering(Extra2, Catagories, CatagoriesNumbered, NoRow);
  //and extract the first several according to number of k
  ExtractFirstNeigbour(Extra2, FirstNeigbour, Catagories, kNumber, k);

  // Calculate the most frequent one appearing
  int cont = 0, cont2 = 0, pos = 0, num = 0, i = 0;
  int Frequency[k], major = 0, posMajor = 0, variable3[k];

  // Initialize the frequency counter
  for(i=0; i<k; i++){
    Frequency[i] = 0;
  }

  //Check repetitions of each number
  for(cont=0; cont<k; cont++){
    num = kNumber[cont];
    pos = cont;

    for(cont2 = 0; cont2<k; cont2++){
      if(kNumber[cont2] == num){
        Frequency[pos]++;
      }
    }
  }

  major = Frequency[0];

  posMajor = 0;

  for(cont=0; cont<k; cont++){
    if(Frequency[cont] > major){
      posMajor = cont;
      major = Frequency[cont];
    }
  }
  //return the numerical catagory that appears the most
  return kNumber[posMajor];
}
////////////////////////////////////////////////////////////////////////////

//calculating Euclidean distance between two points

float EuclideanDistance(float pt1[], float pt2[], int NoAttr){
  int i;
```

```
  float sum = 0;
  for(i=0; i<NoAttr; i++){
    sum = pow(pt1[i] - pt2[i], 2) + sum;
  }
  return sqrt(sum);
}
// ////////////////////////////////////////////////////////////////////////////

//calculation of Euclidean distance between a point and the database

void TotalEuclideanDistance(float pt1[], float variable[], float *Attributes, int NoRow, int NoAttr){
  int i = 0, j = 0;
  float pt2[NoAttr];

  for(i = 0; i<NoRow; i++){
    for(j=0; j<NoAttr; j++){
      pt2[j] = Attributes[i*NoAttr+j];
    }

    variable[i] = EuclideanDistance(pt1, pt2, NoAttr);
  }
}
// ////////////////////////////////////////////////////////////////////////////

//Sorting distances Ascending

void Ordering(float Data[], int Catagories[], int clasesNo[], int NoRow){
  int i =1, j = 1, f = 1, temp[2];

  //  Create a copy of the original classes
  for(i=0; i<NoRow; i++){
    Catagories[i] = clasesNo[i];
  }

  //Ordering
  for(i=1; (i<=NoRow)&&f; i++){
    f = 0;
    for(j=0; j<(NoRow-1); j++){
      if(Data[j+1] < Data[j]){ // descending order >, ascending order <
        temp[0] = Data[j];
        temp[1] = Catagories[j];
        Data[j] = Data[j+1];
        Catagories[j] = Catagories[j+1];
        Data[j+1] = temp[0];
```

```
          Catagories[j+1] = temp[1];
          f = 1;
        }
      }
    }
}
// /////////////////////////////////////////////////////////////////////////////

//Extracting the first N

void ExtractFirstNeigbour(float Data[], float FirstNeigbour[], int Catagories[], int kNumber[], int k){
  for(int i=0; i<k; i++){
    FirstNeigbour[i] = Data[i];
    kNumber[i] = Catagories[i];

  }
}
// /////////////////////////////////////////////////////////////////////////////

//Calculate the most common type: Fashion

int NeighbourFrequency(int ClassifiedArray[], int ElementsInArray){

  int cont = 0, cont2 = 0, pos = 0, num = 0, i = 0;
  int Frequency[ElementsInArray], major = 0, posMajor = 0, variable2[ElementsInArray];

  //  Initialize the frequency counter
  for(i=0; i<ElementsInArray; i++){
    Frequency[i] = 0;
  }

  // Check repetitions of each number
  for(cont=0; cont<ElementsInArray; cont++){
    num = ClassifiedArray[cont];
    pos = cont;

    for(cont2 = 0; cont2<ElementsInArray; cont2++){
      if(ClassifiedArray[cont2] == num){
        Frequency[pos]++;
      }
    }
  }

  major = variable2[0];
  posMajor = 0;
```

```

  for(cont=0; cont<ElementsInArray; cont++){
    if(variable2[cont] > major){
      posMajor = cont;
      major = Frequency[cont];
    }
  }
  return ClassifiedArray[posMajor];
}
//
    ///////////////////////////////////////////////////////////////////////////
```

Listing D.3: Hi-Fi prototype code: k-NN classifier library.h

```
#ifndef kNNFunctions_H
#define kNNFunctions_H

#include <Arduino.h>
//functions for the computation algorithm k-NN


//calculating Euclidean distance between two points

float EuclideanDistance(float pt1[], float pt2[], int co);
//
    ///////////////////////////////////////////////////////////////////////////

//calculation of Euclidean distance between a point and the database

void TotalEuclideanDistance(float pt1[], float variable[], float *
    Attributes, int NoRow, int co);
//
    ///////////////////////////////////////////////////////////////////////////

//Sorting distances Ascending preserving class

void Ordering(float Data[], int clases[], int clasesNo[], int NoRow);
//
    ///////////////////////////////////////////////////////////////////////////

//Extracting the first N

void ExtractFirstNeigbour(float Data[], float FirstNeigbour[], int
    clases[], int kClases[], int k);
//
    ///////////////////////////////////////////////////////////////////////////

```

```
//Calculate the most common type: Fashion

int NeighbourFrequency(int clases[], int k);
////////////////////////////////////////////////////////////////////////////

//Maps a class integer value to a string

char* IntegerToString(int claseint);
////////////////////////////////////////////////////////////////////////////

//Calculate the class of a set of attributes using KNN

int kNNClassifier(float *Data, int clasesNum[], float AccData[], int
    k, int NoRow, int co);
////////////////////////////////////////////////////////////////////////////

#endif
```

Listing D.4: Hi-Fi prototype code: data set example for the data gathered by the author

```

//Database containing the reference dataset, this data is used as the
    "neighbours"

const int NoRow = 100; //  Number of rows in the database
const int NoAttr = 3; //  Number of attributes per row

// Classes alphanumeric database
char* Catagories[] = {"Upright", "Forward", "Backward", "Lying down",
    "Moving"};

//  Attributes: training set
char* AttributeNames[]={"X", "Y", "Z"};

//Data points
float Attributes[NoRow][NoAttr] = {
{2640, -336, 17836},
{2768, -520, 18112},
{4188, -388, 17220},
{2772, -420, 17724},
{4316, -436, 17584},
{4564, -656, 16980},
{4276, -436, 17544},
{5100, -88, 17732},
{4996, -172, 17032},
{4280, -440, 17824},
```

```
{5044, 600, 17192},
{4692, -1488, 17288},
{4692, -188, 17056},
{4968, -1064, 16868},
{5608, -632, 17236},
{5668, -40, 16880},
{5452, -676, 17504},
{5656, -540, 17036},
{4680, -244, 17456},
{4084, -380, 17408},

{-1148, 584, 17592},
{-1556, 692, 17812},
{-1372, 632, 17608},
{-1560, 740, 17692},
{-1420, 652, 17776},
{-1152, 528, 17896},
{-996, 388, 17716},
{-1288, 524, 17740},
{-1724, 540, 17872},
{-1312, 764, 17608},
{-1396, 784, 17668},
{-1672, 684, 17664},
{-1432, 996, 17624},
{-1400, 720, 17688},
{-1220, 664, 17768},
{-952, 632, 17696},
{-1212, 552, 17796},
{-1340, 716, 17896},
{-1228, 708, 17776},
{-1152, 724, 17660},

{10688, -616, 14140},
{11036, -572, 14376},
{10524, -592, 14212},
{9944, -696, 14120},
{10544, -784, 14564},
{10132, -472, 14476},
{10464, -428, 14420},
{10420, -432, 14384},
{10752, -576, 14224},
{10524, -592, 14312},
{10524, -528, 14448},
{10708, -448, 14328},
{10288, -688, 14012},
{10232, -512, 14336},
{9880, -608, 14468},
{10228, -548, 14756},
{10188, -448, 14596},
{10264, -576, 14696},
{10380, -508, 14640},
```

```
{10708, −456, 14320},

{16468, −424, 4532},
{16448, −652, 4400},
{16712, −648, 4044},
{16860, −232, 4344},
{16568, −252, 4508},
{16576, −268, 4568},
{16852, −76, 4520},
{16748, −300, 4632},
{16544, −544, 4340},
{16508, −408, 4888},
{16504, −340, 4660},
{16724, 40, 4844},
{16524, −244, 4704},
{16604, −248, 4848},
{16372, −152, 4852},
{16836, −460, 5000},
{16848, −340, 4672},
{16460, −352, 5024},
{16376, −416, 4892},
{16728, −288, 4536},

{−1444, 4820, 18788},
{2600, 880, 16316},
{−1464, 3800, 17664},
{3724, 5040, 30352},
{1288, −2600, 204},
{2256, 3300, 19996},
{1572, −68, 2876},
{−21816, 668, 8340},
{−1720, −2412, 12176},
{−6968, 13240, 16708},
{−776, 6780, 18312},
{−20276, −10180, 17820},
{−2072, −236, 17248},
{−15656, 20584, 8192},
{2644, −2180, 6104},
{−7400, 5388, 20532},
{−14716, 4732, 32767},
{−14864, 8200, 9472},
{−184, 636, 12280},
{−984, 1516, 5276},

};

//catagory for each data point, done in numbers for memory size reduction
int CatagoriesNumbered[NoRow] = {
0,
0,
```

126	0 ,
127	0 ,
128	0 ,
129	0 ,
130	0 ,
131	0 ,
132	0 ,
133	0 ,
134	0 ,
135	0 ,
136	0 ,
137	0 ,
138	0 ,
139	0 ,
140	0 ,
141	0 ,
142	0 ,
143	0 ,
144	1 ,
145	1 ,
146	1 ,
147	1 ,
148	1 ,
149	1 ,
150	1 ,
151	1 ,
152	1 ,
153	1 ,
154	1 ,
155	1 ,
156	1 ,
157	1 ,
158	1 ,
159	1 ,
160	1 ,
161	1 ,
162	1 ,
163	1 ,
164	2 ,
165	2 ,
166	2 ,
167	2 ,
168	2 ,
169	2 ,
170	2 ,
171	2 ,
172	2 ,
173	2 ,
174	2 ,
175	2 ,
176	2 ,

```
2 ,
2 ,
2 ,
2 ,
2 ,
2 ,
2 ,
3 ,
3 ,
3 ,
3 ,
3 ,
3 ,
3 ,
3 ,
3 ,
3 ,
3 ,
3 ,
3 ,
3 ,
3 ,
3 ,
3 ,
3 ,
3 ,
3 ,
4 ,
4 ,
4 ,
4 ,
4 ,
4 ,
4 ,
4 ,
4 ,
4 ,
4 ,
4 ,
4 ,
4 ,
4 ,
4 ,
4 ,
4 ,
4 ,
4};
```

E APPENDIX

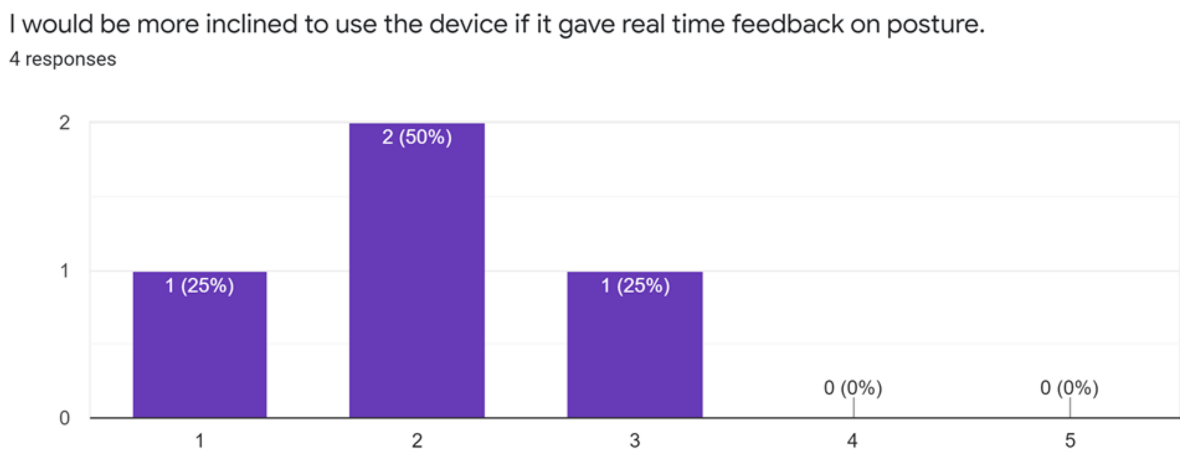

Figure E.1: Version: A, Scale: Strongly agree - Strongly disagree

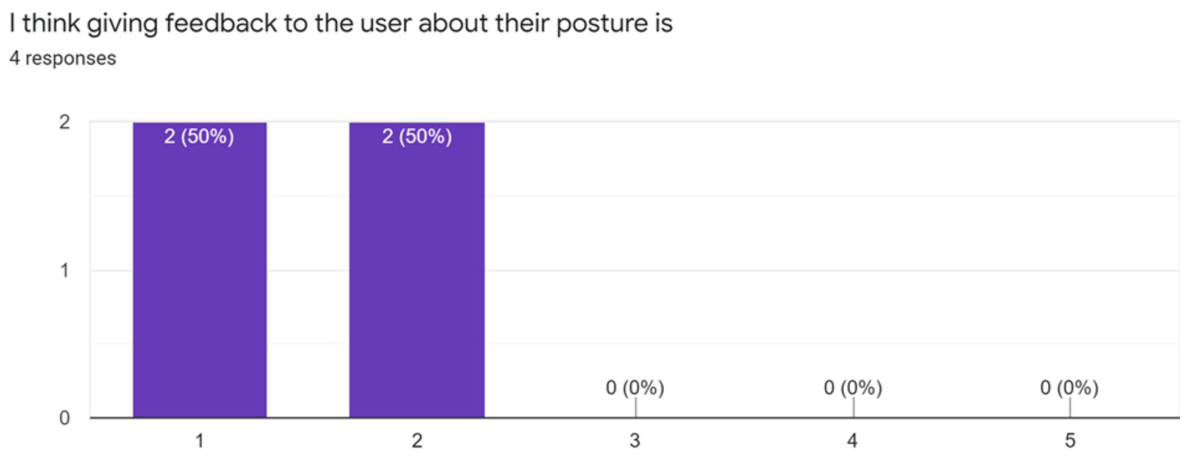

Figure E.2: Version: A, Scale: Very important - Unimportant

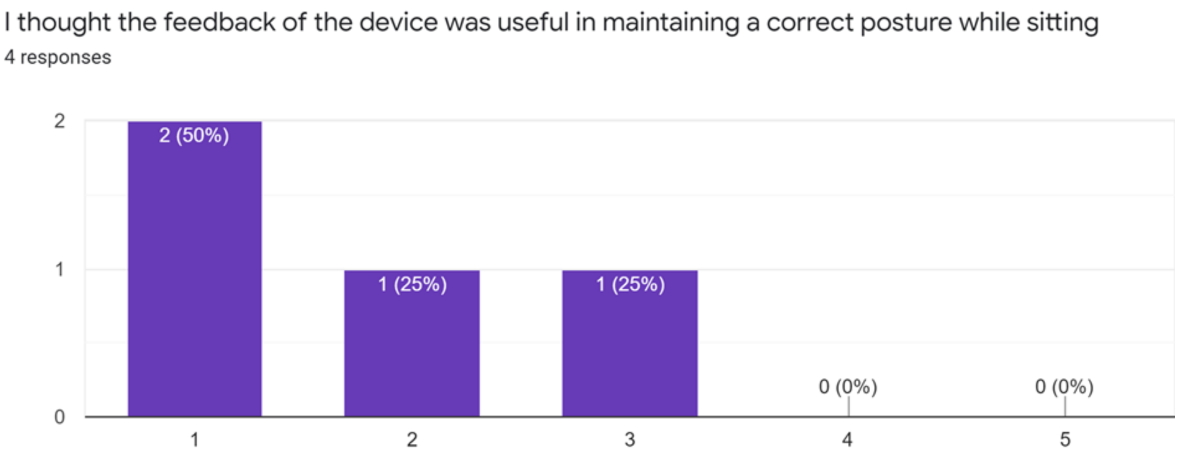

Figure E.3: Version: B, Scale: Strongly agree - Strongly disagree

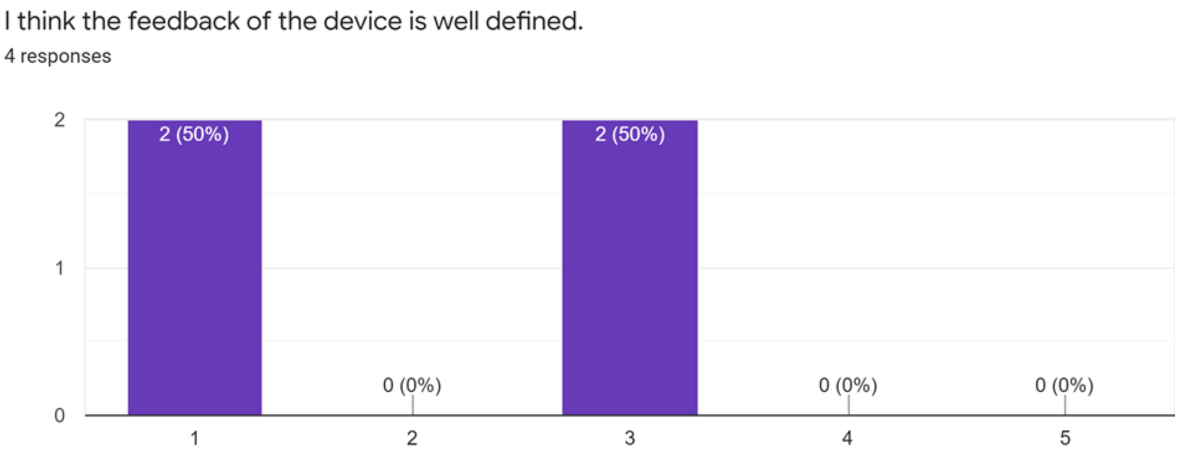

Figure E.4: Version: B, Scale: Strongly agree - Strongly disagree

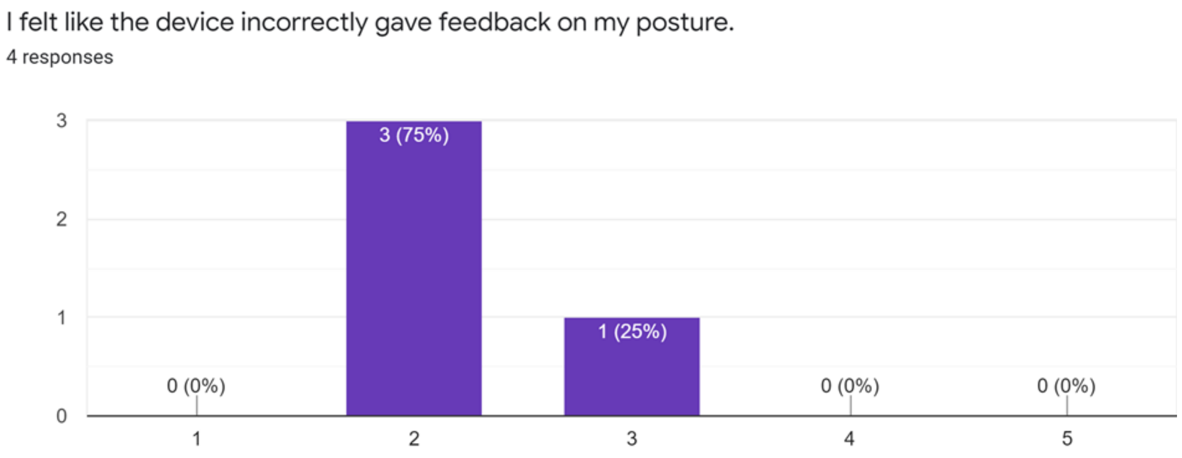

Figure E.5: Version: B, Scale: Definitely - Definitely not

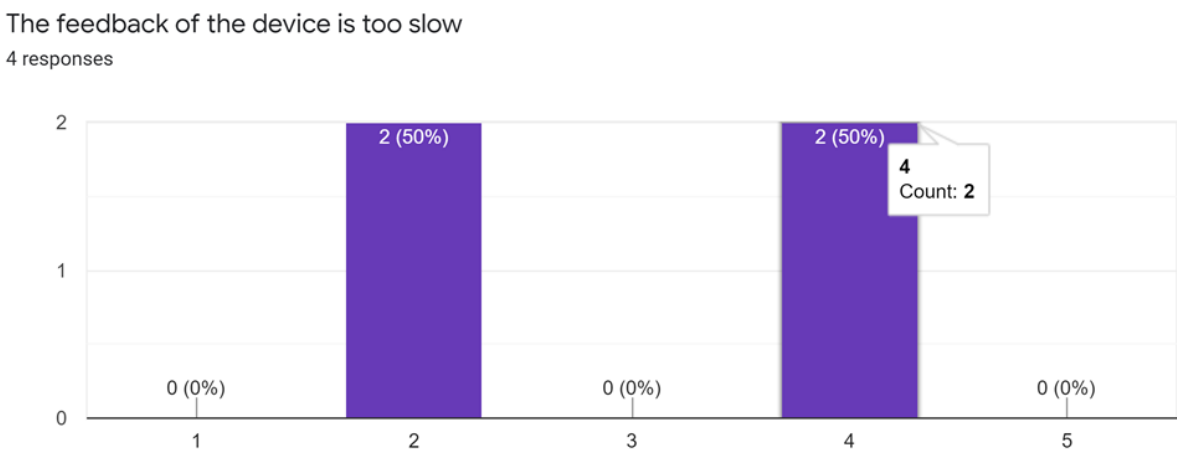

Figure E.6: Version: B, Scale: Strongly agree - Strongly disagree

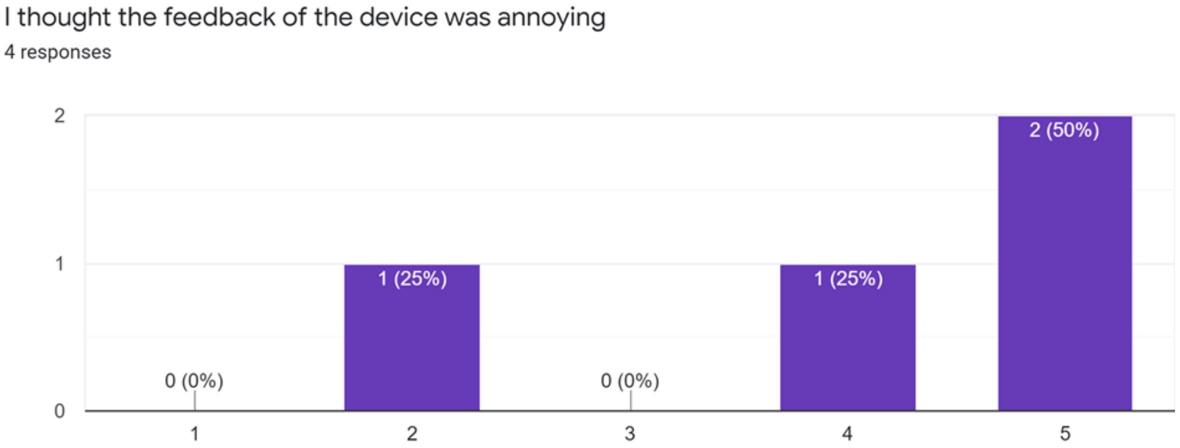

Figure E.7: Version: B, Scale: Strongly agree - Strongly disagree

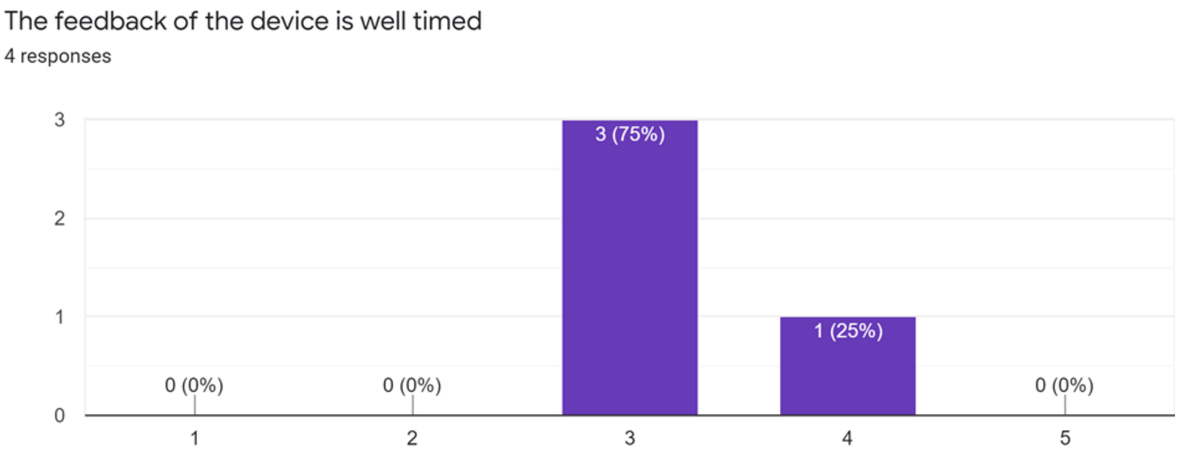

Figure E.8: Version: B, Scale: Strongly agree - Strongly disagree

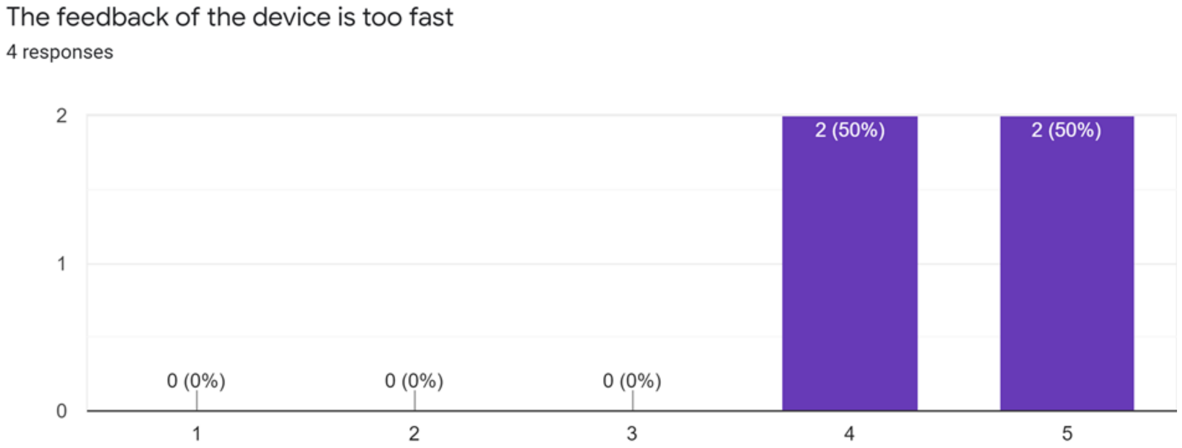

Figure E.9: Version: B, Scale: Strongly agree - Strongly disagree

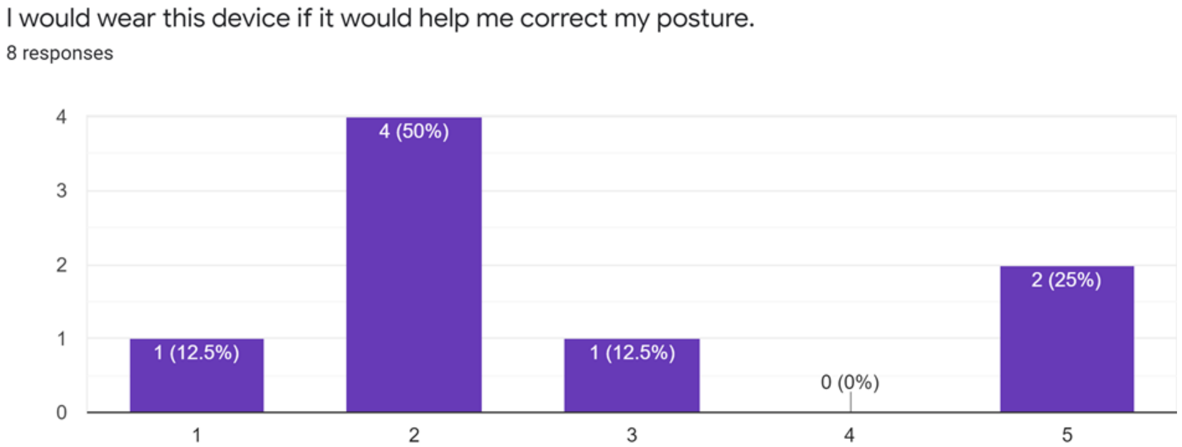

Figure E.10: Version: A&B, Scale: Definitely - Definitely not

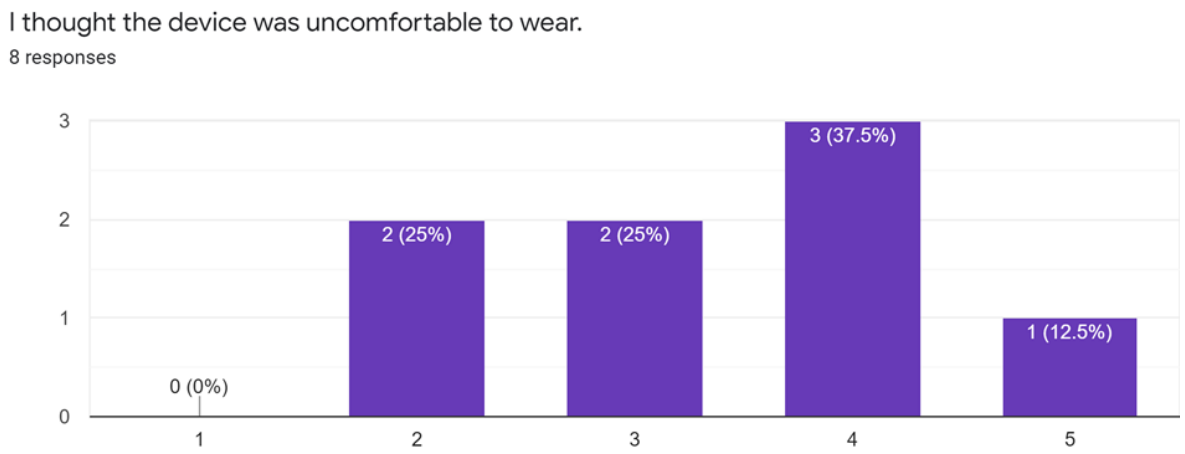

Figure E.11: Version: A&B, Scale: Strongly agree - Strongly disagree

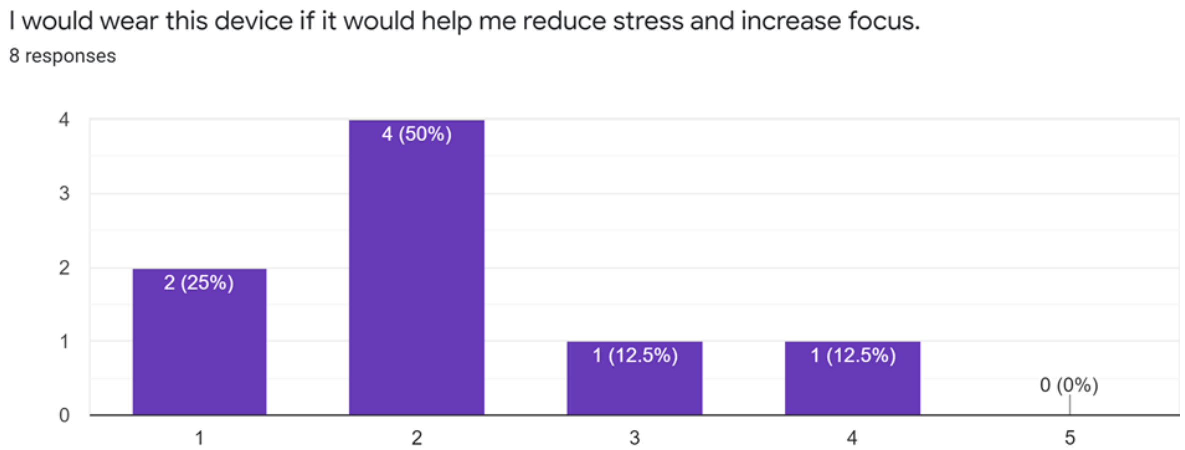

Figure E.12: Version: A&B, Scale: Definitely - Definitely not

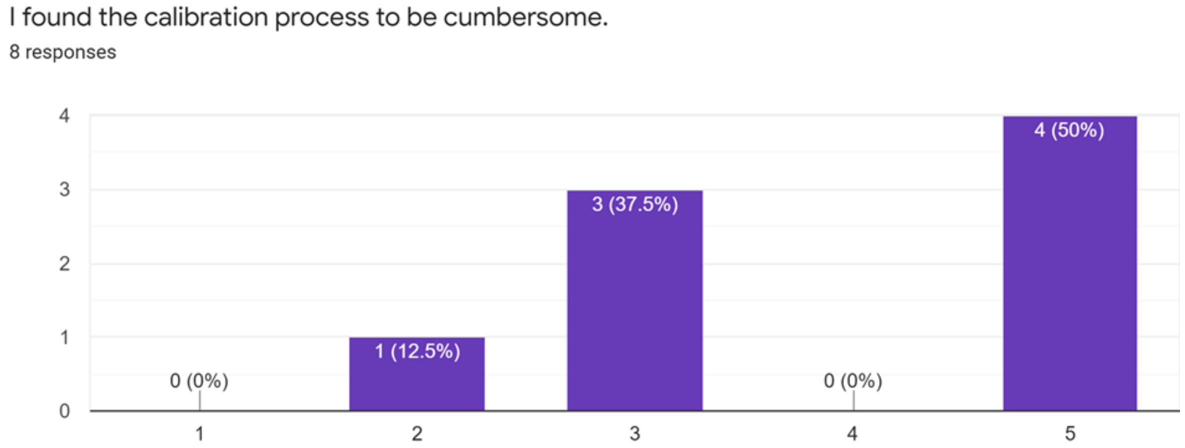

Figure E.13: Version: A&B, Scale: Strongly agree - Strongly disagree

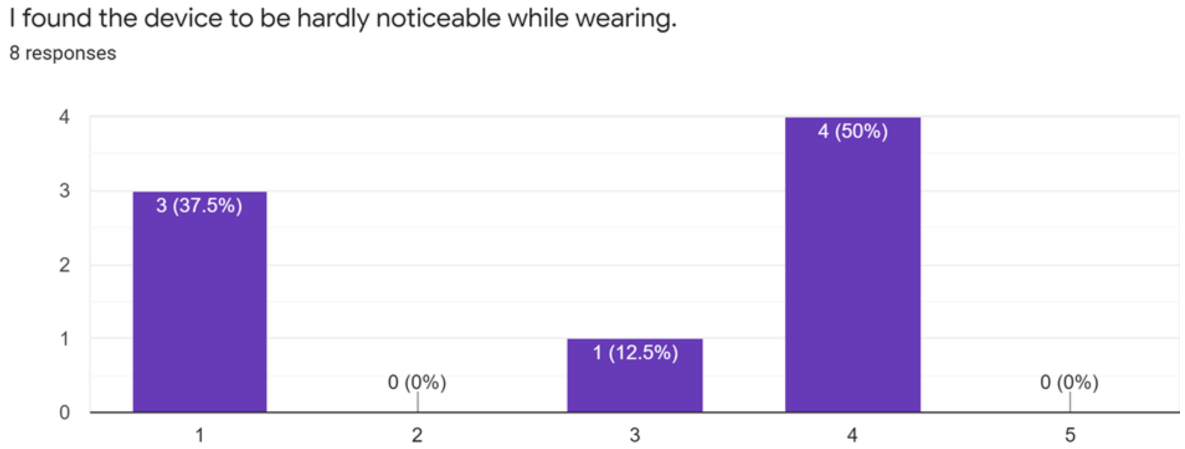

Figure E.14: Version: A&B, Scale: Strongly agree - Strongly disagree

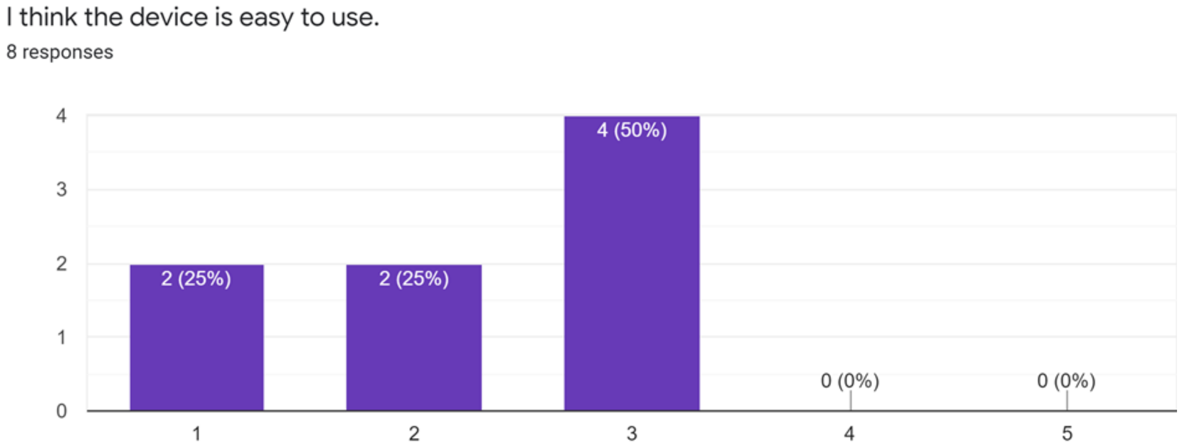

Figure E.15: Version: A&B, Scale: Strongly agree - Strongly disagree

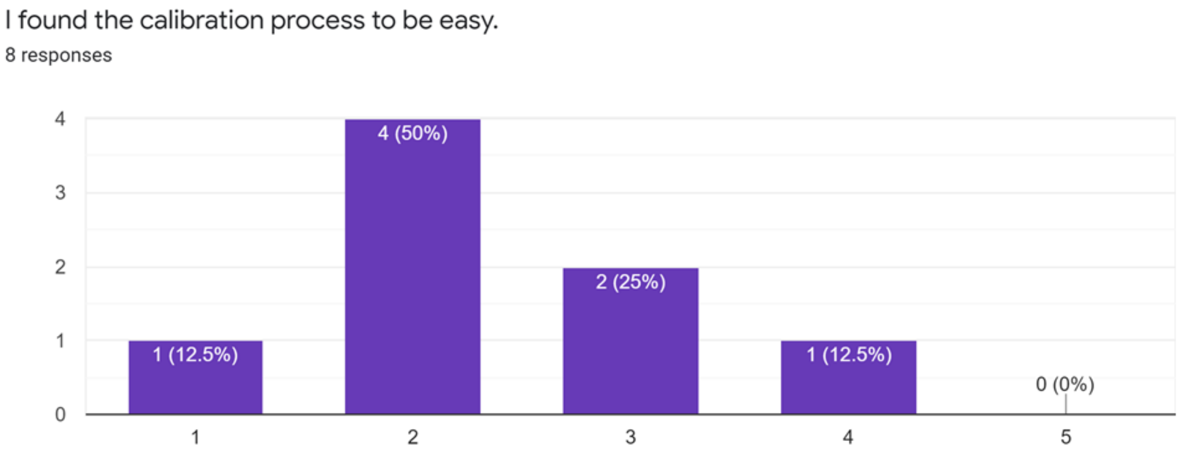

Figure E.16: Version: A&B, Scale: Strongly agree - Strongly disagree

F APPENDIX

INFORMATION SHEET – Breathline
Enschede 01-06-2020

Introduction
This research is conducted for the Breathline bachelor thesis conducted by Martijn Poot a creative technology student. Please take some time to reflect on everything that is in this information sheet you are free to chose not to participate if you wish to. If you do not understand any part of this information sheet or have any other related questions please feel free to ask the researcher. It is also always allowed to ask questions during the research period or afterwards.

Purpose of the research
The research questions this research is helping to answer is the following: "How can a wearable mounted in the umbilical region be used to detect and classify body position, posture and movement data?" The purpose of this experiment is to gather data with the prototype of the device in order to test its accuracy and reliability in identifying posture (sitting upright), body position (standing or sitting and laying down), and movement (walking). Breathing data is also collected to help answer the sub research question: "What benefits can be gained from comparing current breathing pattern to the ideal breathing pattern when identifying sitting position?" where the collected breathing data and posture, body position and movement data are put side by side in order to find correlations.

Type of research intervention
This research will consist of two parts, firstly the device will be worn for around a 1 hour period, during this period you are free to do the normal daily tasks that you have planned except for leaving the house. During this period, the researcher will observe and take note of your posture, body positions and movements in order to compare this to the data collected by the wearable device. In the second part the researcher will ask questions about the usage of the device and your opinion on the device.

Participation selection
You have been chosen to participate in this user test because you are available to the researcher and there is no additional chance of spreading covid-19 by your participation in this research.

Voluntary Participation
Participation in the research is voluntary and you are free to opt out at any moment if you feel uncomfortable or are no longer willing to participate for any reason whatsoever.

Procedures
You will be asked to wear the Breathline prototype device and answer a few questions about the device and its intended usages.

Duration
This research will take 2 hours maximum of which a 1 hour wearing period and the remaining time for introducing the device and questions afterwards.

Risks
Possible risks of this device include electrical shocks from the device or the band connecting it, however because of the low voltage this will not be possible unless the device or band contacts an open wound. Please inform the researcher in you have any open wounds.

Benefits
Participation will benefit the researcher and the development of the Breathline wearable.

Reimbursements
Eternal gratitude may be received upon completion.

Confidentiality
All data collected in this research will be stored securely and anonymized apart from gender and age.

Sharing the Results
The anonymized results might be shared through publication of the bachelor thesis of the researcher. The data will no be shared in any other way.

Right to Refuse or Withdraw
You have the right to withdraw your data from the research if you wish to do so, all data used in the research before the withdrawal of consent can still be used in the research. From the moment consent is withdrawn no new data will be gathered or processed.

Who to Contact
In there are any additional questions or remarks, please contact Martijn Poot via email m.d.poot@student.utwente.nl or call 0031 641643454.

Independent contact
In case you wish to get in touch with an independent person for information about participation in this research please contact DR.IR. E.J. Faber: e.j.faber@utwente.nl.

Rights as a research participant
If you have questions about your rights as a research participant, or wish to obtain information, ask questions, or discuss any concerns about this study with someone other than the researcher(s), please contact the Secretary of the Ethics Committee of the department of EEMCS, drs. Petri de Willigen, mail: ethics-comm-ewi@utwente.nl.

Privacy
All personally identifiable information collected during this research (for example signed consent) will be processed according to the General Data Protection Regulation (GDPR) which can be found at: http://ec.europa.eu/justice/data-protection/re-form/files/regulation_oj_en.pdf.

G APPENDIX

Consent Form for Breathline

YOU WILL BE GIVEN A COPY OF THIS INFORMED CONSENT FORM

Please tick the appropriate boxes	Yes	No
Taking part in the study		
I have read and understood the study information dated 01/06/2020, or it has been read to me. I have been able to ask questions about the study and my questions have been answered to my satisfaction.	○	○
I consent voluntarily to be a participant in this study and understand that I can refuse to answer questions and I can withdraw from the study at any time, without having to give a reason.	○	○
I understand that taking part in the study involves wearing a device for a period of maximum 2 hours and a researcher taking note of posture, movements, and body position for the duration that the device is worn. And that the research asks questions about the usability and comfort of the device.	○	○
Risks associated with participating in the study		
I understand that taking part in the study involves the following risks: physical discomfort of the device being attached around the body.	○	○
Use of the information in the study		
I understand that information I provide will be anonymised and used for the Breathline bachelor thesis or other related publications, and that any pictures taken can used in the Breathline bachelor thesis.	○	○
I understand that personal information collected about me that can identify me, such as [e.g. my name or where I live], will not be shared beyond the study team.	○	○
I agree that my information can be quoted in research outputs	○	○

Signatures

______________________ ______________________ ________

Name of participant Signature Date

I have accurately read out the information sheet to the potential participant and, to the best of my ability, ensured that the participant understands to what they are freely consenting.

______Martijn Poot_____ ___________________ ________

Researcher name Signature Date

UNIVERSITY OF TWENTE.