Analysis of Sleeping Patterns Using Smartphone Sensors

Semester Thesis

Steven Meliopoulos
Suhel Sheikh

Distributed Computing Group
Computer Engineering and Networks Laboratory
ETH Zürich

Supervisors:

Johannes Schneider
Prof. Dr. Roger Wattenhofer

December 10, 2011

Abstract

Current research suggests that many people suffer from symptoms caused by sleep deprivation due to poor sleep hygiene or sleeping disorders. In today's fast-paced world, getting enough rest out of one's sleep is especially important.

Various alarm clocks based on sleep cycle evaluation as well as software implementations on smartphones, making use of the built-in accelerometer, are available. Their main goal is to detect the optimal moment for waking up by using collected activity data. However, opinions on their efficacy are mixed and there is no supporting scientific data. Furthermore it might be more effective to get to the root of the problem by making people aware of unhealthy habits. One way of achieving this is by presenting easily understandable statistics which make it simple for the users to recognize patterns and adjust their behavior accordingly. While most of the available solutions offer some statistics, they lack reference values which makes the data hard to interpret.

Our target was to build an application for smartphones which not only includes the aforementioned intelligent alarm functionality but also improved statistics containing global reference values. We therefore explored existing research on the topic of sleep analysis based on a person's movements during sleep. In order to assess how suitable smartphones are for recording movements we conducted several experiments and derived an algorithm that is used by our application to extract the user's motions from the phone's accelerometer signal. In addition we enhanced our system by adding a server that collects the data from all users enabling them to compare their sleep to other users directly on their phone or via a website.

Contents

1 Motivation and Existing Solutions

1.1 Sleep and Sleep Disorders

The right amount and quality of sleep is essential to human well-being. While the exact purpose of sleep is not completely known yet, it is clear that various disruptions of sleeping patterns can have serious implications.

It has been shown that sleep deprivation has adverse effects on the immune system and the memory. According to estimates about 40% of the adult population in western cultures is affected by problems with falling asleep or daytime sleepiness [1]. Despite their large prevalence and substantial consequences, sleep disorders tend to be ignored by individuals and society as a whole. The symptoms are often just accepted and dealt with - possibly because they are not considered acute or because they appear slowly and gradually. Even just obtaining a diagnosis - for example using the methods mentioned above - is often deemed too tedious.

There exist two physiologically different types of sleep referred to as REM and non-REM or NREM sleep respectively. REM sleep is characterized by rapid and random eye-movement, high activity of the brain but paralysis of the body. It is believed that most dreaming takes place during REM sleep and that it is particularly important for processing memories. The paralysis is viewed as a protection mechanism to prevent movements that occur in dreams from being acted out. NREM sleep is currently further divided into the stages NREM 1-3 (previously 1-4). During NREM sleep the eyes move very little and the different stages are also characterized by distinct electroencephalograms. N1 is the lightest and N3 the deepest form of sleep, meaning that the least amount of stimulation is required to awaken a person from stage N1 as compared to N2 and N3. In the deeper stages there is almost no muscle activity and people often feel groggy for some time if they are awakened during deep sleep.

During a night the REM and NREM stages normally occur in multiple cycles in the order N1 $\rightarrow$ N2 $\rightarrow$ N3 $\rightarrow$ N2 $\rightarrow$ N1 $\rightarrow$ REM. The period lies between 90 and 120 minutes. REM sleep usually makes up between 20 and 25 percent of the total sleep duration while the REM phases are longer at the beginning and become shorter towards the end of the night. The relative amount of REM sleep varies between people and is also influenced by age, sleep disorders, drug intake, and other factors. [2]

Common methods for determining the progression of sleep stages in subjects include polysomnography and actigraphy. Polysomnography refers to the comprehensive recording of the biophysiological signals that characterize the different stages of sleep such as brain waves, heart rate, eye movement,

and muscle activation. From the resulting polysomnogram (PSG) a trained expert can then infer the current sleep stage for every epoch - typically segments of 30 seconds - by applying a set of well established rules. This process is referred to as sleep stage scoring or sleep staging. Because all the relevant signals are captured, polysomnography is regarded as the gold standard against which all other methods are compared.

In actigraphy only the movement of the subject is recorded. This is accomplished by using an actigraph, which is a small device containing acceleration sensors that is worn like a watch around the wrist or the ankle by the patient. Actigraphy is mostly used to distinguish only between sleep and awake states. Using actigraphy for sleep stage scoring is rather difficult as it produces a much smaller amount of data than polysomnography but a number of algorithms have been proposed that promise reasonable results. Instead of actigraphs, some of these methods rely on special mattresses that are designed to detect movements more precisely. However a major advantage of actigraphy lies in the fact that it is much less intrusive: While polysomnography can only be carried out in a sleep laboratory where multiple sensors need to be attached to the patient, which makes it expensive and which might also interfere with the results, using actigraphy the patient can sleep at home in a familiar environment. This makes actigraphy especially suitable for large field studies. The two methods can also be applied in combination and are both used for diagnosing sleep disorders.

1.2 Existing Solutions

In recent years a number of consumer products have appeared that aim at solving sleep related problems. Most of them focus on difficulties with waking up in the morning, which many people share. In the following we will shortly present two especially popular variants.

1.2.1 Standalone Implementations

There are a number of devices on the market promising to help users waking up in the morning. A notable example is “aXbo”. “aXbo” is an alarm clock system that is designed to awaken the user at an optimal moment. It consists of a device similar to a traditional alarm clock and a sensor bracelet that is worn around the wrist by the user during the night. The bracelet measures the user’s movements and sends the data wirelessly to the clock. The user specifies the time by which he needs to be awake. During a certain duration leading up to that time the alarm clock will then try to determine the user’s current sleep stage using the data provided by the bracelet and ring the alarm when the user’s sleep is the least deep. According to the vendor this will make the process of waking up as comfortable as possible and thus the

user will feel refreshed and well rested in the morning and throughout the whole day.

1.2.2 Smartphone Software

Modern smartphones have become extremely widespread in the past few years. Apart from high mobility they offer universal capabilities comparable to personal computers as well as a number of special sensors. Platforms such as Android, iOS and Windows Phone make it simple to develop and distribute applications to a wide user base even if their hardware varies a lot. These ideal properties probably explain the great number of sleeping aid applications that are available today. One of the first and most successful example of this kind is an application for iOS which is called "Sleep Cycle", developed by "Maciek Drejak Labs AB". It is described to work much in the same way as "aXbo". But instead of wearing a bracelet, the user has to place the phone on the bed during the night. The application uses the phone's inbuilt accelerometer to measure the user's movements which are conveyed to it via the mattress. While standalone implementations such as "aXbo" can often be rather expensive, with smartphone applications the risk involved with buying is very small for the user, as these apps are usually very cheap or even free.

1.2.3 Criticism

While the promises made by these products sound very appealing, there are some problems with the described approach:

- Being awakened during light sleep is certainly desirable but it is unclear whether and how strongly this will affect how one feels throughout the whole day.
- The algorithms used by these products are proprietary and thus kept secret. It is therefore impossible to tell whether they are actually able to identify different sleep phases correctly and reliably. Customer feedback is partially very positive but because there is no concrete evidence for these systems' effectiveness, it is possible that this is only due to a placebo effect.
- An easy wake up process alone does very little towards making the user aware of possible problems with his sleep habits.

1.3 Goals

Our goal for this thesis is to develop a system to help users identify possible sleep problems. Because of all the advantages described above we intend

to implement it as an application for smartphones, namely Windows Phone devices. We want to explore to what extent sleep actigraphy using smartphone accelerometers is possible and develop algorithms to extract useful information about the sleep quality from the collected data. Using this information, an alarm function similar to the existing solutions can then be included but in addition the information should be displayed to the user in an accessible manner that enables him to identify unhealthy patterns in his behavior. To lend more meaning to the obtained statistics it should also be possible to collect data from all users so that global reference values can be provided.

2 Data Gathering

In order to construct our algorithm we had to collect raw actigraphy measurements of different test persons over the course of multiple nights.

2.1 Test Device

For these tests we used Google Nexus One devices running the Android operating system instead of the Windows Phone device on which we would develop the final application. The first reason for this was that for security reasons the Windows Phone 7 operating system provides no convenient way of exporting application data, which would make the analysis of the recordings impractical. Furthermore we had four Nexus One devices at our disposal as opposed to only one Windows Phone device. Therefore using the Nexus One for our experiments allowed us to record measurements with multiple identical devices at the same time which proved to be helpful. We found that the two types of devices and operating systems were sufficiently similar so that our findings from one platform could be applied to the other.

2.2 Sensing and Recording

Practically all modern smartphones contain a sensor module which measures the acceleration that acts on the device along the three spatial axes separately. The Nexus One phone, which we used for our tests, contains a Bosch BMA150 module, which is capable of sensing accelerations as small as 0.04m/s^2[3]. The exact components are not published for all devices but we expect them to be similar. The measured values are made available to application developers in the form of events. Such an event contains the current reading for all three axes as floating point numbers. In our tests these events arrived on average approximately 25 times per second with a standard deviation of 2. This allowed us to view the duration between events as constant in order to simplify our algorithms.

We created a simple Android application that captures all of these events and writes the raw measurement values together with a timestamp directly to a file, which could later be transferred to a computer for examination.

2.3 Device Placement

Different measurement results for placing the device on the left or right side of the test person are certainly to be expected because limb movements of the

opposite side will be dampened more strongly compared to movements on the side the device is placed. However we were interested in the differences of placing the measurement device at different vertical positions in bed. Our objective was to determine whether the user has to adhere to a strict positioning of his device and if so which positioning will allow us to detect most of the user's movements.

2.3.1 Experimental Setup

To analyze variations of measurement in different locations in bed we used four Google Nexus One smtartphones running our raw data gathering application. The device positions were numbered from top (1) to bottom (4).

Figure 2.1: Experimental setup of devices in bed to determine the optimal position for recording

2.3.2 Results

We repeated this experiment three times for the duration of a whole night each. The results of our measurements can be seen in table 2.1. We did not find a connection between the device's position and the number of detected events. For each measurement a different position detected the highest number of events, which suggests that there is no optimal vertical device placement position. Our measurements indicate that our algorithm (see

Number of detected movements				
	measurement 1	measurement 2	measurement 3	total
position 1	26	24	26	76
position 2	21	27	28	76
position 3	19[a]	25	24	68
position 4	19	29	23	71

[a]This value has been interpolated since the recording was incomplete.

Table 2.1: Overview of device placement measurement results

section 3.2) does not depend on the user placing his device in a pre-defined position.

2.4 Measurement Reproducibility

In order to determine how reproducible our measurements are we compared the accelerometer measurements of two devices which were tightly sticked together using adhesive tape, thereby ensuring that virtually the same acceleration acted on both devices throughout the experiment. The experiment was carried out over the course of a whole night and lasted 470 minutes. The raw accelerometer recordings, of which the absolute values are shown in figure 2.2, showed similarities when comparing the relative position of peaks, but differed greatly in terms of their absolute value and also individual peak heights. After we applied our algorithm to the recordings of both devices we could observe that the results had an acceptable agreement of 47%. The output of our algorithm can be seen in figure 2.3. In total there were 17 events throughout the whole night, excluding movements at the beginning and end of the experiment which were caused by placing the devices and stopping the recording. Out of these 17 events 8 were detected by both phones, while 9 events were only detected by one of the two phones.

Detected events			
	exclusive events	shared events	percentage of total events
phone 1	5	8	76.47%
phone 2	4	8	70.59%

Table 2.2: Overview of reproducibility measurement results

The probability of coincidentally producing at least one random matching event on the second device in this experiment can easily be calculated considering that the experiment lasted 470 minutes, which we can divide into 28200 buckets of length 1 second. Matching one of m timestamps recorded by the first device with one of n (randomly produced) timestamps on the

second device can be expressed using the following equation, where t is the number of possible timestamp buckets:

$$p = 1 - \prod_{i=0}^{n-1} \frac{t - m - i}{t} \tag{2.1}$$

$$p = 1 - \prod_{i=0}^{11} \frac{28200 - 13 - i}{28200} = 0.78\% \tag{2.2}$$

Replacing m, n and t with the actual values (2.2) of our experiment (m=12, n=13, t=28200) shows that it is indeed very unlikely that the high correlation of the two devices is accidental. We conclude that our measurements have a good reproducibility.

2.5 Heart Rate Measurement

During our experiments we also recorded the heart rate as a reference signal because the measurement equipment is easily available and inexpensive. We experimented with two types of devices that are both capable of recording their measurements throughout a whole night and transferring the data to a computer for further analysis. One was a system as it is often used in sports, consisting of a wrist watch and a chest strap that contains the sensor and sends its measurements wirelessly to the watch, which records them. The second one was a finger clip that additionally measures the oxygen concentration in the blood. The latter type is more commonly used in medical applications. However both devices occasionally suffered from the problem that they would lose contact with the body during the night and not be able to produce measurements anymore.

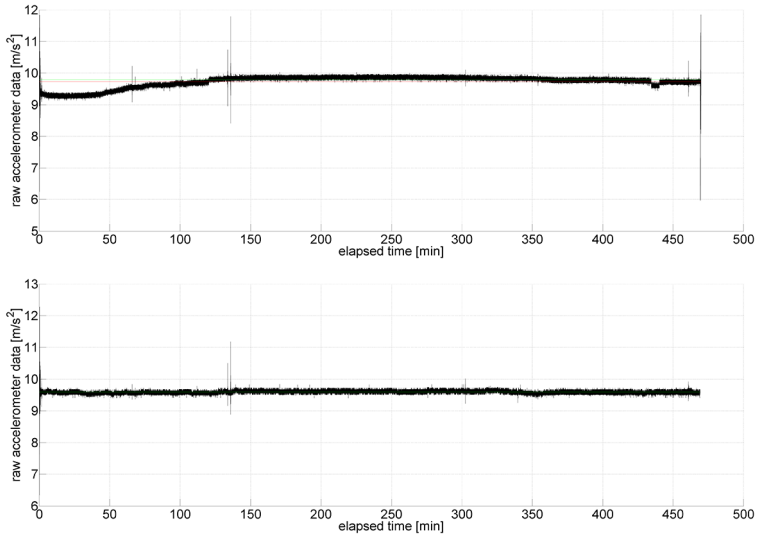

Figure 2.2: Raw recording of phone 1 and 2

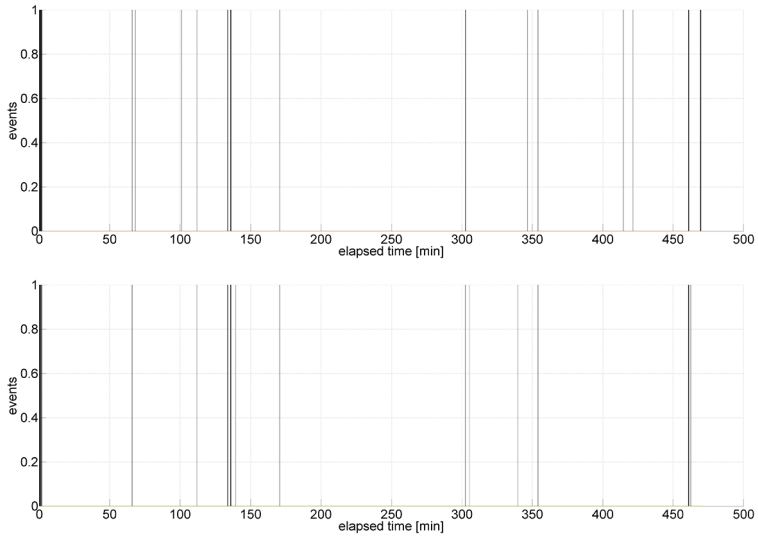

Figure 2.3: Events detected by our algorithm in recording of phone 1 and 2

3 Algorithm

3.1 Related Work

As mentioned earlier, polysomnography is the most reliable method for sleep stage scoring available today. It does however suffer from a number of drawbacks. To obtain a recording the subject needs to spend one or multiple nights in a specialized sleep laboratory and multiple observatory devices and electrodes need to be attached to the patient. This results in high cost for equipment and manpower. In addition to this the results might be biased due to the method's invasiveness and because the subject might be uncomfortable sleeping under such unfamiliar circumstances. The analysis of the recorded data is performed by hand by a specialist who assigns a sleep stage to every 30 second interval throughout the whole night. Aside from being tedious and time consuming, this process also introduces subjectivity. For these reasons a lot of research has gone into developing algorithms that can be used to perform automatic sleep stage scoring based on actigraphy data alone.

Jansen and Shankar developed a method for sleep staging using movement related signals obtained from a static charge sensitive bed (SCSB)[4]. The SCSB was proposed as a noninvasive and precise method for recording body movements during sleep in 1979 by Alihanka and Vaahtoranta[5]. In a SCSB there are two metal plates beneath the mattress that are separated by a stiff insulator. Movements of the subject sleeping in the bed can be detected by observing changes in the potential between the two plates. Jansen and Shankar obtained six recordings of the SCSB signal from six volunteers. Simultaneously they recorded the electroencephalography, electromyography, and electrooculography signals to be used as a reference. They extracted fifteen features from each 30 second epoch of the the SCSB signal such as the number of zero crossings of the signal and its derivatives and the signal bandwidth. Half of the data set was then used to determine which of these features are good indicators of different sleep stages by applying stepwise discriminatory analysis. By then using these discriminators on the other half of the test set they achieved a rate as high as 89% for distinguishing between the stages awake, REM and non-REM.

In a study conducted in 2008, Tilmanne et al. compared four algorithms for distinguishing sleep and wake states [6]. Instead of recording their own data they used a large existing database of recordings from infants. The recordings consist of a full PSG and actigraphy signals recorded with bracelets worn around the ankle by the infants instead of a SCSB. Two of the algorithms they examined are so called linear combination methods while

the other two are classified as pattern recognition methods. The principle of the linear combination methods, namely Sadeh's algorithm and Sozonov's algorithm, is to define a discriminant function that is a linear combination of different features extracted from the actigraphy signals, similar to the method by Jansen and Shankar described above. To adapt the parameters of the linear combination to their data set they minimized the sum of squared errors. The pattern recognition techniques that were examined are artificial neural networks (ANNs) and decision trees. ANNs aim at replicating the functionality of biological neural networks. They are composed of multiple interconnected nodes or neurons which produce an output depending on the weighted sum of their inputs, typically described by a sigmoid function. The neurons are grouped into several layers which are arranged in series, with the first layer representing the input and the last layer the output. All the neurons of a given layer process the input in parallel. In a decision tree, the input is fed to the root node and at each internal node a decision is made, that partitions the output space until a leaf is reached. A leaf represents the final classification, which in this case is either "awake" or "asleep". These pattern recognition methods are deemed especially appropriate for the task at hand because they have been shown to be effective for modeling nonlinear input-output relationships. The parameters of the ANN and the decision tree are also trained by minimizing the squared sum of errors between their output and the reference PSG. The four methods achieved accuracy rates between 78.7% and 82.1% with the two pattern recognition methods scoring slightly higher than the linear combination methods.

The results of these studies suggest that sleep staging based solely on actigrahy data is possible with a satisfying degree of accuracy. The obvious course of action for us therefore seemed to be to select and implement one of these well tested algorithms to enable our application to recognize different sleep stages. Unfortunately however this proved impracticable for the following reasons: Each one of the presented methods relies on adapting its parameters as well as testing its accuracy by using a reference PSG. The cost of using a sleep laboratory to create a sample data set and having the PSGs scored by an expert would have highly exceeded the limits of a semester thesis. Using existing data was not a viable option either because any actigraphy data recorded either with a SCSB or bracelet actigraphs is expected to be incomparable to ours, due to our very different recording hardware and technique. We therefore proceeded by attempting to independently develop a method for extracting as much useful information from our recorded signals as possible.

3.2 Filtering

Our recording application produces CSV files that we transferred to a computer and analyzed using Matlab. Figure 3.1 shows the plots of the raw data we recorded during a normal night along the three axes as well as the absolute value that we computed from them. The most noticeable feature they exhibit are the abrupt jumps accompanied by strong peaks, especially well visible on the y-axis plot. In between these jumps the curve stays relatively flat while exhibiting high frequency noise in the order of about 0.15m/s^2. The jumps can be explained by shifts in the position of the phone resulting in changed amounts of gravitational acceleration acting on each axis. The peaks represent the movements that caused these shifts. This assumption is supported by the fact that the jumps disappear almost completely in the plot of the absolute acceleration while the peaks remain.

To confirm our assumptions so far and to be able to characterize the noise more precisely we created a recording that should show no signs of movement at all by placing the phone on a stable flat surface for over ten hours. The resulting plot is shown in figure 3.2. At the first glance the plot seems to exhibit the same jumps but this is only due to the different scale. It does however confirm that the constant small changes in the reading are in fact caused by noise. Another interesting property is the ascending slope during the first 140 minutes. We found the reason for this to be that the phone was plugged in to a power supply and 140 minutes was the time it took for the battery to be fully charged. This scenario is very likely to occur in normal usage of the final applications because users would not want their battery to be empty in the morning, causing the alarm not to go off. Therefore we had to take this effect into consideration.

The conclusion from these findings was that we are not interested in the general level of the curve over a period of time but only in momentary changes. Therefore the first step in filtering the signal would be to take the first derivative or more precisely the difference between adjacent elements. So if we view the sequence raw sensor values as a vector

$$\vec{a}_{\text{raw}} = [a_{\text{raw},0}, a_{\text{raw},1}, \ldots, a_{\text{raw},n}]$$

then after this first step it would become

$$\begin{aligned} \vec{a}_{\text{s1}} &= [a_{\text{s1},0}, a_{\text{s1},1}, \ldots, a_{\text{s1},n-1}] \\ &= [a_{\text{raw},1} - a_{\text{raw},0}, a_{\text{raw},2} - a_{\text{raw},1}, \ldots, a_{\text{raw},n-1} - a_{\text{raw},n}] \end{aligned}$$

Furthermore we decided that we would concentrate only on the absolute value because it contains all the events and distinguishing between events that occur along different axes does not yield any useful additional information.

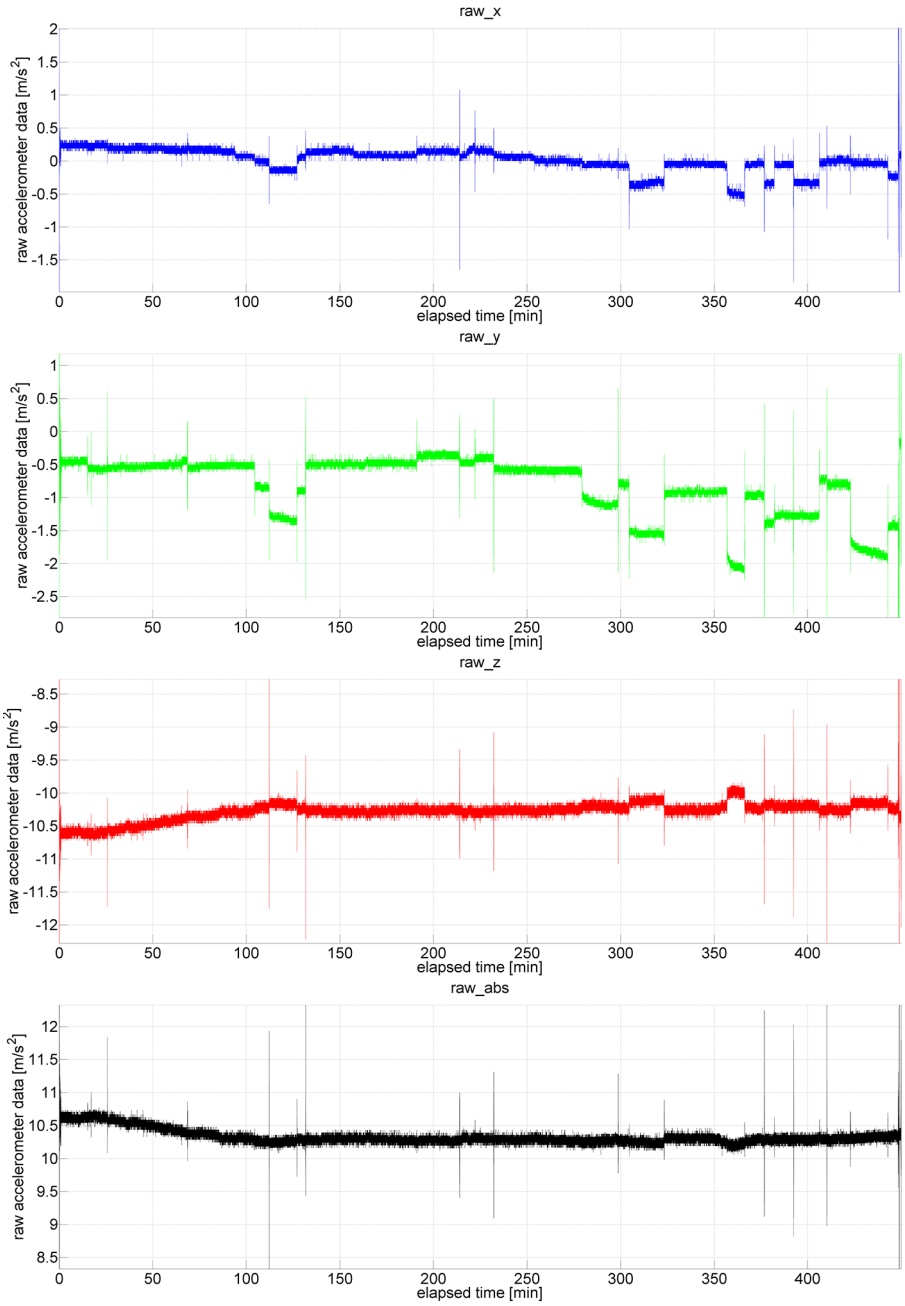

Figure 3.1: Raw recording of a normal night for x, y, and z axes and absolute value

Figure 3.3 and figure 3.4 show what the plots look like after differentiation. To eliminate all the noise we applied a threshold t to cut off all the values whose absolute value is below the maximum found in the empty

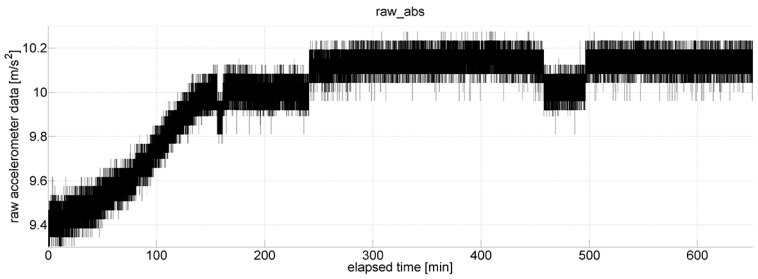

Figure 3.2: Raw recording of no movement at all exhibiting the characteristic shift caused by charging the phone

recording $\vec{e}$.

$$t = \max(|e_i|) \; \forall e_i \in \vec{e}$$

During our tests we used a threshold value that we determined statically. Since we expect the amount of noise to vary with different devices the final application would have to include a calibration function to measure the threshold value on each device individually.

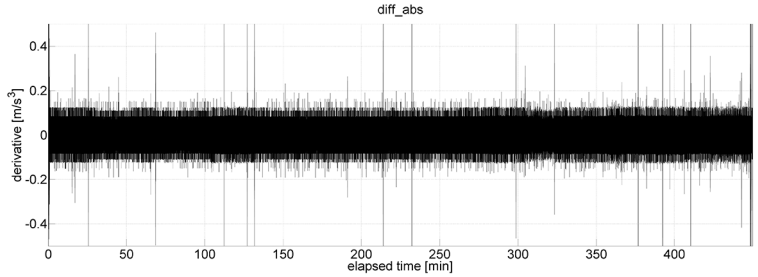

Figure 3.3: First derivative of absolute value for a normal night

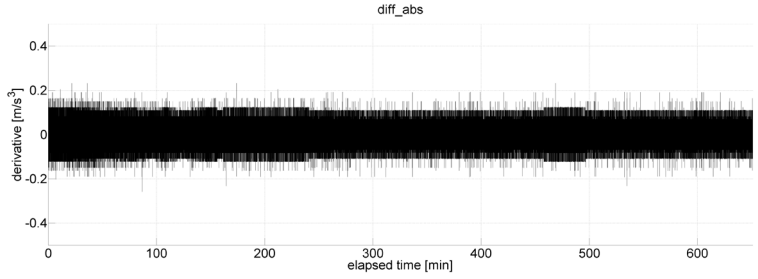

Figure 3.4: First derivative of absolute value of recording without movement

As mentioned in section 2.4, the height of the peak for a given event can differ depending on the recording device. Moreover, the recorded intensity of a movement also depends on how far away from the phone it occurs: If the

user moves a foot at the diagonally opposite side of the bed from the phone this event will have a much weaker impact than if a hand is moved directly next to the phone, even though both events are equally important. For this reason we decided to view events in a binary way: either they happen or they don't but we don't consider their intensity. Therefore the elements $a_{s2,i}$ of our new vector are now

$$a_{\mathrm{s2},i} = \begin{cases} 1, & |a_{\mathrm{s1},i}| > t \\ 0, & \text{else} \end{cases}$$

In a final step we would assign to each minute of the recording the total number of events that occurred during that minute. Figure 3.5 shows the result of this procedure. This form of the data then serves as the basis for all further analysis. The recording of the heart rate during the same night is shown in figure 3.6. It can be seen that elevations in the heart rate occur simultaneously with the peaks in the graph of the filtered movement data which suggests that the extracted information is indeed meaningful.

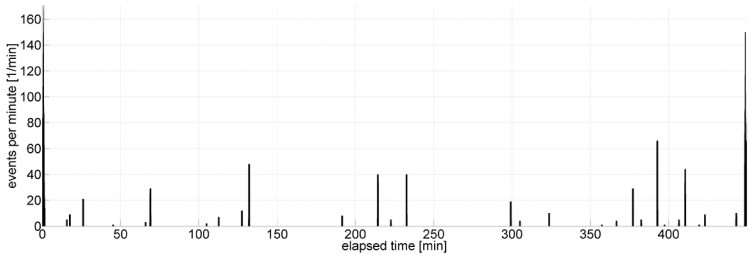

Figure 3.5: Final result of filtering

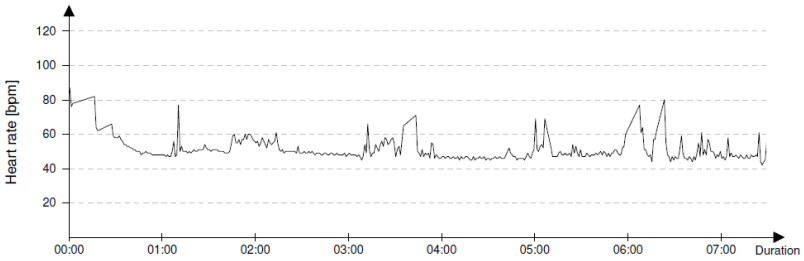

Figure 3.6: Heart rate during test night

4 Client Implementation

This chapter discusses the implementation of our sleep analysis smartphone application.

4.1 Platform

The Windows Phone Application Platform allows the use of the Silverlight and XNA application frameworks. While XNA provides a high performance framework, which is mainly targeted at game development, Silverlight is focused on facilitating the development of applications making use of Windows Phone's Metro style UI elements. Given that our application is not performance-critical we decided to develop our application using the Silverlight framework.

4.2 Goals

The main goal of our application is to provide the user with useful and correct information about his sleep. The information is presented in such a manner that the user understands its meaning and that he can not only detect deviations from previously recorded nights but is also made aware of differences compared to the average user of the application.

Our application implements the filtering algorithm derived in section 3.2 and presents the user a graphical representation of his movements, helping him to spot phases of great agitation. This can help the user to identify possible external disturbances during his sleep, such as for example the bell-ringing of an adjacent church or it can visualize the effects of internal influences, such as alcohol, on the user's sleeping behavior. The application enables the user to carry out all sorts of experiments and monitor their effects on his sleep.

All recorded nights are stored on the user's device and can easily be accessed at all times. Stored data must contain all important information but at the same time be small enough so that a large number of recordings can be stored. In order to generate the global reference values every recording is submitted to our server (see chapter 5) if an Internet connection is available.

The application shall not detect in which sleep phase a user is, as is claimed by various other smartphone applications. Due to the reasons explained in section 3.1 we decided that attempting to guess the sleep stages without a verifiable algorithm would not yield any useful results and so we

limited ourselves to displaying only correct and objective information that is still open to the user's interpretation.

4.3 Calibration

Our filtering algorithm depends on knowing the maximum derivative of the accelerometer sensor's noise floor, which we found varied among different devices. An important consideration in the design of the calibration function was that while a more elaborate calibration algorithm could possibly result in more precise results we need to keep in mind that the calibration process should be as simple as possible, limiting the possibilities for errors. The final implementation asks the user to place the phone on a completely still surface and confirm when he is ready to start the calibration. We then monitor the maximum value during 20 seconds. If the user picks up the phone while the calibration is still running we recognize the intense movement, automatically stop the calibration process and inform the user that the calibration failed because the phone has been moved. During our tests we found that 20 seconds are enough to reliably detect the sensor's maximum noise derivative. In 50 tests our calibration algorithm did not fail to detect the correct value once.

4.4 Filtering and Recording

Probably the most important characteristic of our implementation of the algorithm derived in section 3.2 is its very low memory footprint and its optimization for good performance on a smartphone. While all our computations using Matlab were operating on the complete night's raw accelerometer data, including all three axes, this approach was not feasible for use on a mobile device. Recording raw accelerometer data for all three axes generated files of 5-6 megabytes per hour and therefore requiring approximately 45 megabytes for a full night's recording. Not only did the high memory consumption make this approach infeasible for use on a mobile phone, but also the computation time needed for running these large files through our filtering algorithm. The user expects to be presented the information on his recorded night immediately without having to wait a minute or even longer for our computations to finish.

We decided to adapt our filtering algorithm to operate simultaneously to recording accelerometer values by using a buffer. This change allowed us to drastically reduce the memory footprint by 99.3% on average resulting in file sizes of under 1 kilobyte.

Our adapted filtering algorithm is automatically passed new accelerometer values after registering itself as an event listener for the accelerometer object's ReadingChanged event. By using this event as an entry to call our

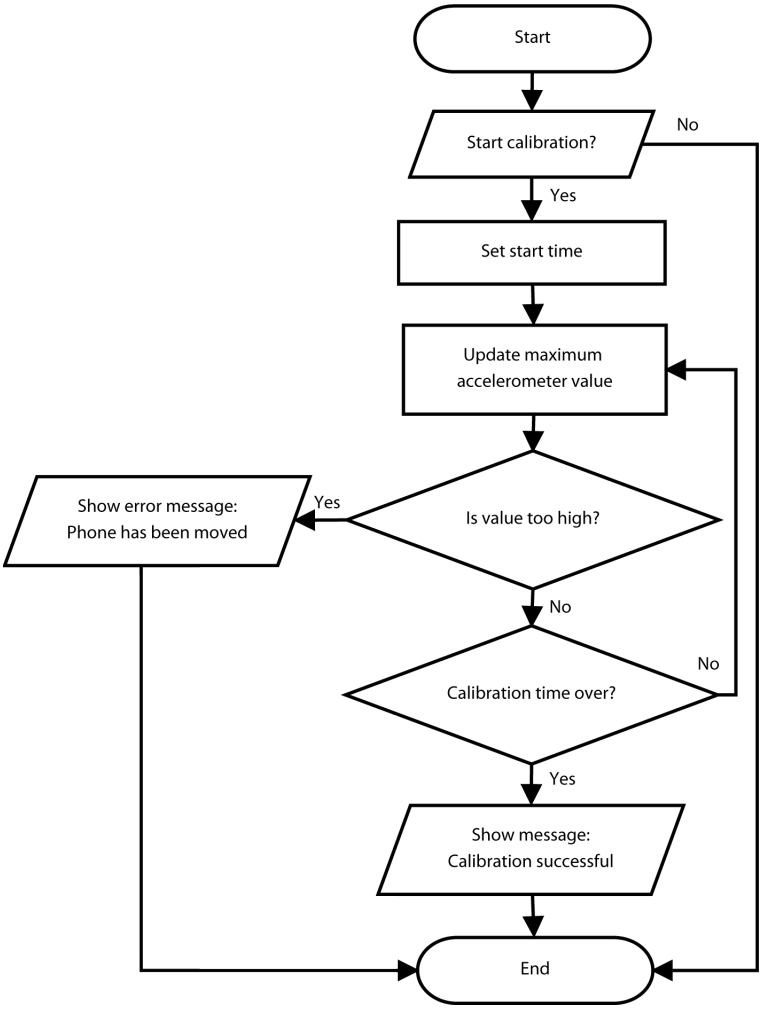

Figure 4.1: Calibration process flowchart

filtering function we ensure that it is only called when new values have actually arrived. Since the accelerometer events are generated at a very high rate we must ensure that our event handling function is as light-weight as possible. We therefore decided to generate a simple recording function which counts how many events it has detected in a given minute. A flowchart describing the outline of our recording event handler can be seen in figure 4.2. In order to prevent data loss in case the phone powers off due to low battery charge level or our application gets terminated for any other reason we write all data to our application's isolated storage as soon as it has been recorded.

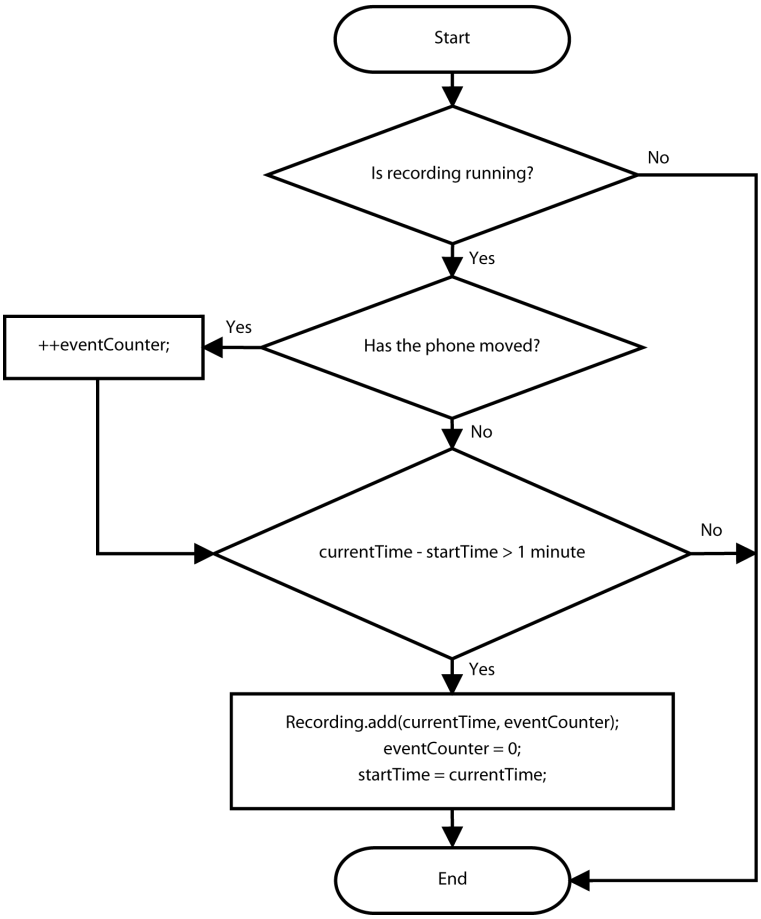

Figure 4.2: Recording flowchart

4.5 Sleep Chart

It is very important to offer the user a visual representation of his sleep, not only to give him an overview but also to build confidence in our application. Users will trust our application's information if they can verify that our application did indeed register what happened throughout the night. Reading through feedback submitted for similar applications we found a large number of users were putting the application through synthetic test cases, performing checks such as leaving their smartphone completely still on their drawer for a complete night. While it is questionable whether such tests are suitable for determining the correctness of a sleep tracking application it is undoubtedly of great importance that users trust our application and it is therefore imperative that they are given the possibility to verify that our application did actually record their movement and that it is not only presenting made up statistics.

Consequently we chose to display an activity graph illustrating the filtered accelerometer data instead of a graph showing our estimate on the user's sleep-wake state. Showing estimated sleep-wake state information would fail if the user conducts a synthetic experiment as mentioned before and would make him perceive the validity of our estimate as very low no matter how high our average accordance with more elaborate sleep-wake detection methods such as polysomnography is.

4.6 User Interface

The user interface of our application has been designed with Windows Phone's Metro design language in mind. Metro is a typography-based design language, which refrains from using fancy graphics and crowded content in favor of a clear and delightful experience. Windows Phone makes use of Metro throughout the whole operating system, which is often seen as one of the major upsides of the OS by its proponents. We decided to implement our UI using a panorama control. A panorama control is basically a big canvas for an application's content that can be scrolled horizontally, which contrasts to the layout known from traditional mobile applications, which typically only allow vertical scrolling. The main panorama page of our application can be seen in figure 4.3.

Presenting recorded data in a visually pleasing way was of high priority to us. It is crucial for our global online statistics to get as many users to use our application as often as possible. An appealing and fluid UI, besides providing useful functionalities, is key to getting users to like an application. We therefore put much effort into the design of a nice and easy to use UI. The home screen (figure 4.3) offers the user the possibility to set the alarm time without having to browse through multiple pages. Displaying this setting on the start screen instead of the settings page is important because it allows the user to see what time he set the alarm to without having to navigate through multiple pages. Other less frequently used settings, such as a wake up song picker (figure 4.4), can be accessed through the settings page, keeping the application's start page clutter-free. We also added a feedback tile to the start screen which not only contains a counter indicating how long the current recording has been running but which can further change its background color for example to emphasize error messages or to make the user aware that the recording or calibration is running.

4.7 Alarm

Alarm functions used by most sleep tracking devices and applications differ from alarms known from regular alarm clocks. The basic principle behind these intelligent alarms is to use the estimate of the user's current sleep

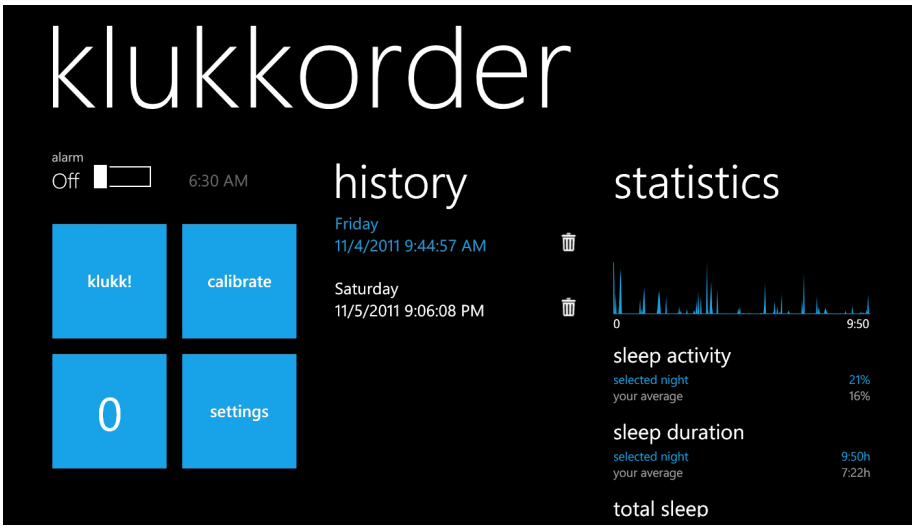

Figure 4.3: Home panorama page of our application

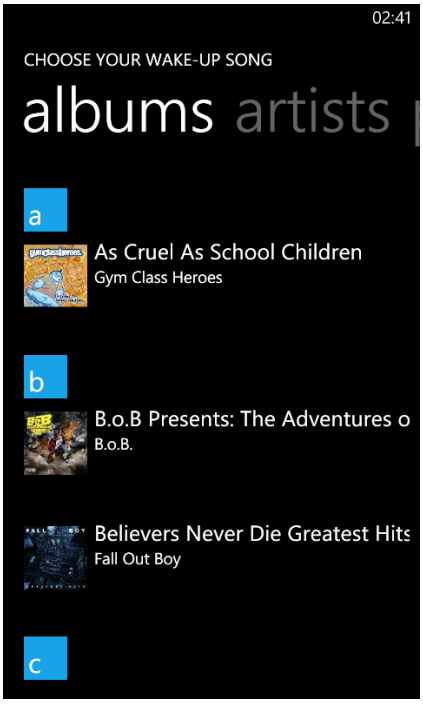

Figure 4.4: Wake up song picker page

phase to decide whether the user would wake up feeling refreshed when woken at this point. As mentioned in section 1.1 it is unfavorable to wake up during a deep sleep phase because the likelihood of feeling drowsy is high in such cases. Given that we know movements to be primarily found in light sleep phases we could develop our alarm such that it wakes users when it estimates them to be in a light sleep phase. The user is asked to enter his desired alarm time and agrees to be awoken within a 30 minute time frame leading up to the alarm time. As soon as our application detects that the current time is within this time frame it starts checking whether the user has been active recently. If the application finds that the user has recently moved it rings the alarm otherwise the application keeps checking for movement until movement is detected. When no movement could be detected the application will ring the alarm once the alarm time has been reached.

5 Server implementation

The second part of our system is the web server. It satisfies two purposes at the same time. Its first task is to receive recordings from the client application running on the users' phones, store them, compare them, and send results back to the client. The second function is to present information about the collected data through a website.

5.1 Platform

The server was implemented using the open source web application framework Django. Although the complexity of the server in its current form is limited, the use of a framework still has a number of advantages. Django offers a convenient abstraction layer for access to the underlying database as well as for handling HTTP requests and the generation of web pages is simplified by a flexible templating system. Besides enabling faster development, using a framework also aids extensibility. For instance if we wanted to add user accounts in the future, this could be achieved without major complications.

5.2 Data Upload

Recordings are uploaded from the client application to the server via simple HTTP requests. The following properties are contained in the request's POST data: the actual recording i.e. a list of minute numbers and their corresponding event count, the time at which the recording was started, its duration, and the user's sex and age. From this information the following additional properties are then computed: the total number of events in the whole recording, the average number of events per minute, the corresponding weekday and moon phase, and the bedtime rounded to one hour. All these properties are then stored in the database together with the time of the upload and the uploader's IP address. Some of these properties are redundant, but included to enable simpler querying and others serve no purpose at the moment but may be used in the future.

After storing the recording the server sends a response back to the client containing some of the computed information which can then be displayed in the application.

5.3 Website

The entry page of the website is meant to provide a quick overview. It displays the five most recently uploaded recordings as graphs. This lets users see what other people's recordings look like so they can compare them with their own. Additionally the page asynchronously polls the server every few seconds for new uploads that might have happened since the page was loaded. If new recordings are available, the server responds with the new data in JSON format and the new plots are immediately created on the page using JavaScript, specifically the open source flot library. This live ticker gives an impression over how actively the service is used. It also serves as a confirmation and incentive for users to see their recordings immediately appear on the website after they upload them.

The main part of the website focuses on identifying possible connections between different properties of the recordings. In the current state of the implementation we focused on the influence of the weekday, bedtime and moon phase on the average event frequency and the sleep duration. All these relationships are represented visually, resulting in six bar graphs. The selected characteristics are however merely a small example of what is possible. After the service has been running for long enough one could for example investigate whether the month of the year had an influence as well and separate statistics could be shown based on age and sex. What is important to note about these statistics is that they are relevant not only to users of the client application but for anybody interested in different influences on sleep quality. They might therefore raise the application's publicity and help it gain a larger user base, which in turn is beneficial to the quality of the collected data.

Just as with the live ticker, the graphs are created on the client side via JavaScript and only the JSON data is loaded from the server. This greatly reduces the load on the server and allows us to always present the most up to date statistics with every request instead of serving cached results that are only updated periodically.

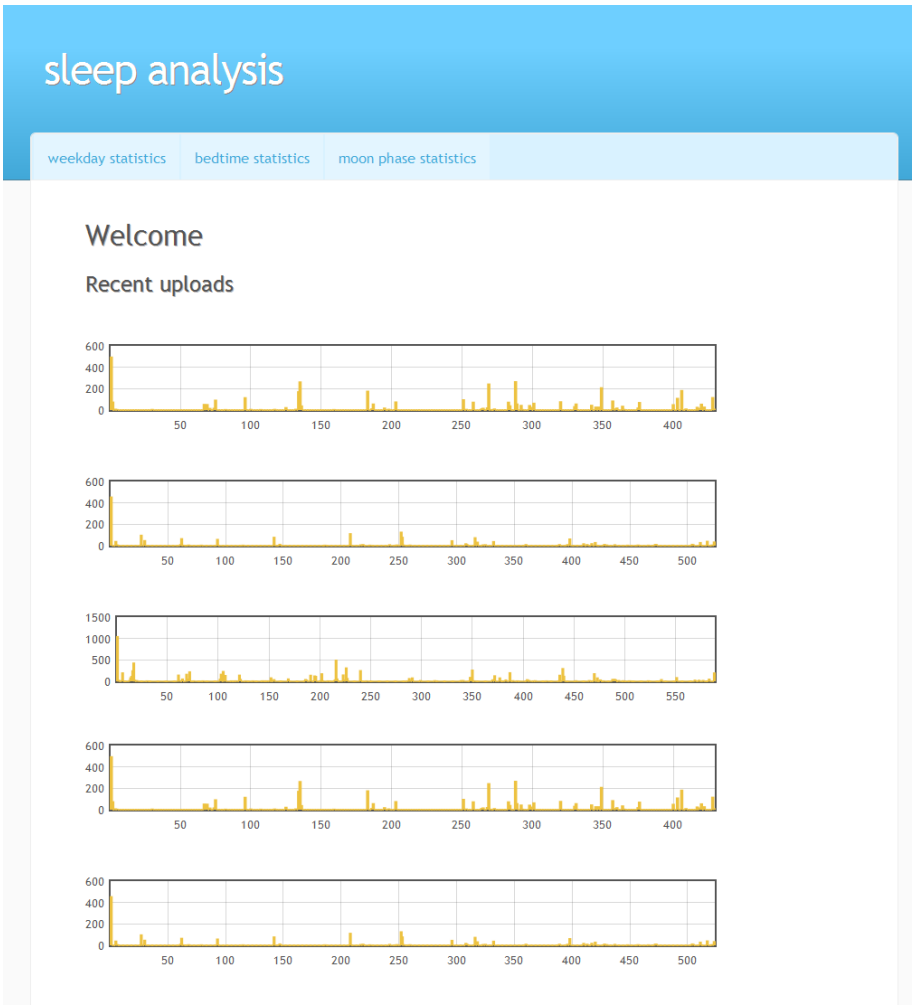

Figure 5.1: Entry page of the website showing the most recent uploads (sample data)

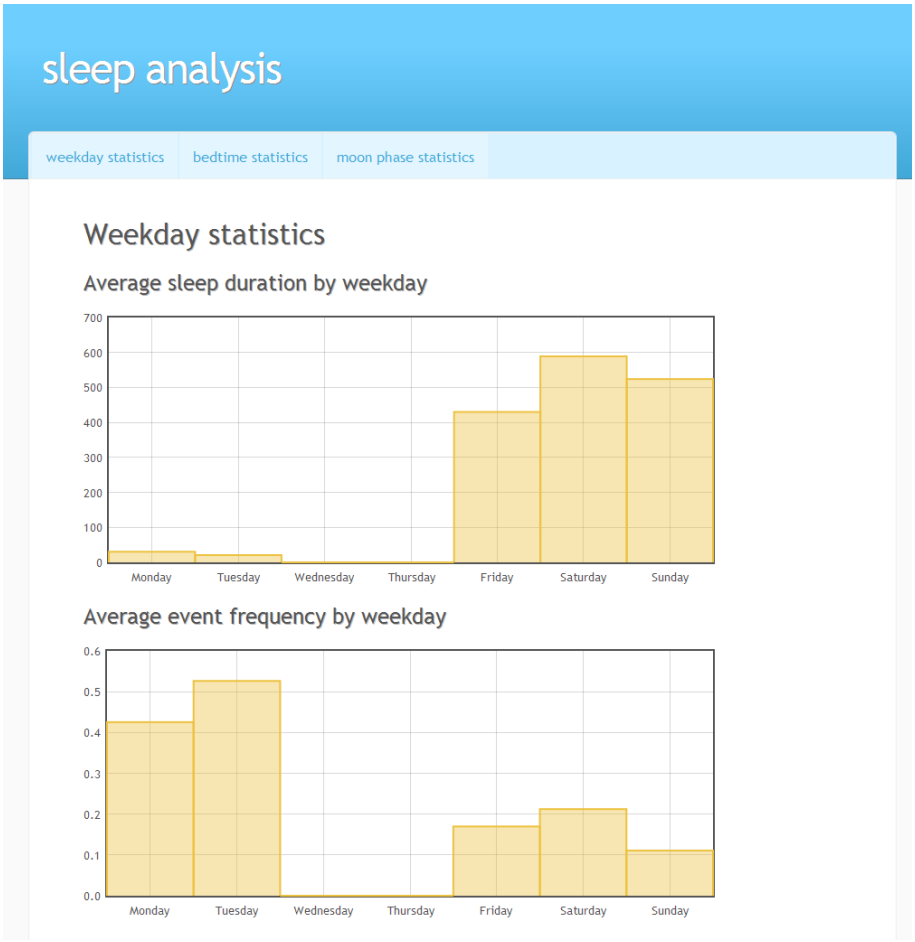

Figure 5.2: Graphs showing the influence of the weekday on sleep duration and event frequency (sample data)

6 Conclusion

The experiments we have conducted have shown that a smartphone placed on the bed next to a sleeping person is able to sense the person's movements even though they might be very gentle and dampened by the mattress. We constructed an algorithm that is able to separate meaningful events from the considerable amount of noise that the phones' sensors produce. The algorithm is designed to ensure that the exact type of hardware of the phone and also its position on the bed do not have an influence on the results. Although sleep staging based on actigraphy alone seems to be feasible, we were not able to adapt an existing method or develop our own because the necessary reference data could not be produced.

We implemented an application for the Windows Phone platform that enables users to create recordings of their movements during their sleep which are then filtered using our algorithm. The recordings are saved on the phone and can be displayed at any time.

Despite not being able to recognize sleep stages we still included a simple implementation of an intelligent alarm clock that tries to avoid waking the user up during deep sleep stages, which are characterized by a very low amount of activity.

Our system also comprises a server that collects the recordings from all the client devices and stores them in a database. The server then provides the users with global averages to which they can compare their own values. In addition to this it presents various statistics about the collected data on a website.

6.1 Future Work

We believe that sleep staging using our recording technique is technically possible. If a large enough number of recordings with reference PSGs in addition to the phone's recordings from different test subjects are obtained, one could apply the methods described in section 3.1 to develop an algorithm suitable for usage in a smartphone application.

We also think that the client-server architecture we implemented makes our system attractive for sleep research. It provides a simple and inexpensive way to collect a large amount of data from subjects sleeping in a realistic environment, which is otherwise always a big challenge. Both the client and the server can be adapted to collect more information from the uses. For example the application could require the user to fill out a survey about his sleeping habits as it is common in sleep studies.

A simple measure to increase the size of the potential user base would be to create implementations of the client application for other popular mobile platforms such as iOS and Android which should be possible with only minor adjustments.

Smartphones also contain other sensors that could be leveraged for the evaluation of sleep quality. For example the microphone could be used to detect snoring or apneas. Signals from different sensors could also be combined to obtain even more detailed insights. The increased power consumption caused by the additional sensors should not be as much of a problem as it is in truly mobile applications since the device can remain plugged in during the night. However the algorithms should still be able to process all the data in real time because otherwise the storage requirements would quickly become too high.

Another interesting addition would be to enable the application to be used by couples sleeping in the same bed. Two phones placed on either side of the bed could communicate with each other in order to establish which movements they should attribute to which side of the bed depending on which device sensed them more strongly.

Bibliography

[1] J. L. Hossain and C. M. Shapiro, "The prevalence, cost implications, and management of sleep disorders: An overview," *Sleep and Breathing*, vol. 6, pp. 85–102, 2002.

[2] M. A. Carskadon and W. C. Dement, *Principles and Practice of Sleep Medicine*. Elsevier Saunders, 4th ed., 2005.

[3] Bosch Sensortec GmbH, Reutlingen, Germany, *BMA150 Data Sheet*.

[4] B. H. Jansen and K. Shankar, "Sleep staging with movement-related signals," *International Journal of Bio-Medical Computing*, vol. 32, pp. 289–297, 1993.

[5] "A static charge sensitive bed. a new method for recording body movements during sleep," *Electroencephalography and Clinical Neurophysiology*, vol. 46, pp. 731–734, 1979.

[6] J. Tilmanne, J. Urbain, M. V. Kothare, A. V. Wouwer, and S. V. Kothare, "Algorithms for sleep-wake identification using actigraphy: a comparative study and new results," *Journal of Sleep Research*, vol. 18, pp. 85–98, 2009.