Extracting and Filtering Data using Layered Architecture

Bjarki Freyr Sveinbjarnarson

Project Report of 30 ECTS credits
Master of Science (M.Sc.) in Computer Science

June 2021

Extracting and Filtering Data using Layered Architecture

Thesis of 30 ECTS credits submitted to the School of Computer Science at Reykjavík University in partial fulfillment of the requirements for the degree of
Master of Science (M.Sc.) in Computer Science

June 2021

Project Report Committee:

Supervisor: Anna Sigríður Íslind. Assistant Professor, School of Computer Science, Reykjavík University, Iceland

Supervisor: Erna Sif Arnardóttir, Assistant Professor, School of Computer Science and School of Engineering, Reykjavík University, Iceland.

Babak Farshchian, External Examiner
Associate Professor, Department of Computer Science, NTNU (Norwegian University of Science and Technology), Norway

María Óskarsdóttir, Internal Examiner
Assistant professor, Department of Computer Science, School of Technology, Iceland

Extracting and Filtering Data using Layered Architecture

Bjarki Freyr Sveinbjarnarson

June 2021

Abstract

Gathering data has always been a mandatory part of research since the fundamental role of every research project rests in data. Therefore, it is vital to quickly overview the data to grasp the gathered data's essence. The Sleep Revolution project will contain more than 10,000 sleep measurements. The purpose of this thesis is to find a method to represent better the collected data, which brings forward the questions: i) How can we design and develop a system to extract data from scored sleep measurements? ii) How can layered architecture be used to read, display and filter the extracted data? A layered architecture is used in this thesis, where a system for extracting information from data is designed and developed. The layered architecture in the designed system makes it: i) more reusable, ii) less fragile for changes, and iii) more adaptable for extensions. The industry has often used extraction and filtering to represent and understand data better. However, prior work from the industry that uses filtering and extraction for data has its work hidden, making it difficult to evaluate for research purposes. In research, attempts either lack the variety or volume of data, or the projects do not develop a software architecture to represent their system adequately. For this thesis, the following tests were conducted: i) Scalability tests for reading and filtering, ii) usability tests, iii) AttrakDiff and System Usability Scale Questionnaires and, iv) semi-structured interviews.

The work for this thesis shows a promising start through the design and development of a layered architecture for displaying and filtering the extracted data collected for research purposes. However, the system needs further work and more extensive testing to understand the value and usability of the designed system in more depth. The main contribution of this thesis is the designed layered architecture, the extraction and display system, and a set of design principles to guide others working on a project with similar traits.

Útdráttur og síun gagna með notkun lagskipts arkitektúrs

Bjarki Freyr Sveinbjarnarson

júní 2021

Útdráttur

Söfnun gagna hefur ávallt verið mikilvægur fylgisteinn rannsókna. Þar af leiðandi er mikilvægt að geta skoðað gögnin á áhrifaríkan hátt. Verkefnið Svefnbyltingin mun innihalda meira en 10.000 svefnmælingar. Þessi ritgerð snýst um að finna aðferðir til að endurspegla gagnagrunninn með því að svara eftirfarandi spurningum: i) Hvernig getum við hannað og þróað kerfi til að vinna úr gögnum skoraðra svefnmælinga? ii) Hvernig er hægt að nota lagskiptan arkitektúr til að lesa, birta og sía skoruðu gögnin? Í þessari ritgerð, þar sem hannað og þróað er kerfi til að vinna upplýsingar úr gögnum, er notast við lagskiptan arkitektúr. Lagskiptur arkitektúr gerir hannaða kerfið: i) endurnýtanlegt, ii) minna viðkvæmt fyrir breytingum og iii) aðlaganlegt fyrir viðbótum. Iðnaðurinn hefur oft notað útdrátt og síun til að tákna og skilja betur gögn. Hins vegar er erfitt að meta og skoða þær aðferðir þar sem sú vinna er oft óaðgengileg. Rannsóknir hins vegar skortir oft fjölbreytni, magn af gögnum eða innihalda ekki arkitektúr til að endurspegla hugbúnaðinn á fullnægjandi hátt. Til að endurspegla almennilega notagildi hugbúnaðins voru eftirfarandi próf gerð: i) stigstærðarpróf fyrir lestur og síun, ii) notagildispróf, iii) spurningalistar AttrakDiff og System Usability Scale og iv) hálfuppbygð (e. semi-structured) viðtöl.

Afurðir verkefnisins sem gerðar voru í kjölfar ritgerðarinnar eru nokkrar. Þar má nefna kerfi til að draga út breytur úr “skoruðum” mælingum, gagnvirkur hugbúnaður sem sýnir útdregnu gögnin og gerir notenda kleyft að sía þau og lagskiptur arkitektúr á hverju útbúnu kerfi. Hins vegar þarf meiri vinnu áður en kerfið getir orðið hjálplegt tól innan Svefnbyltingarinnar. Helsta framlag þessarar ritgerðar er afurðir verkefnisins ásamt hönnunarreglum sem gætu nýst öðrum sem vinna að verkefni með svipaða eiginleika.

Extracting and Filtering Data using Layered Architecture

Bjarki Freyr Sveinbjarnarson

Project report of 30 ECTS credits submitted to the School of Computer Science at Reykjavík University in partial fulfillment of the requirements for the degree of
Master of Science (M.Sc.) in Computer Science

June 2021

Student:

…………………………………………………………………….
Bjarki Freyr Sveinbjarnarson

Project Report Committee:

…………………………………………………………………….
Anna Sigríður Íslind

…………………………………………………………………….
Erna Sif Arnardóttir

…………………………………………………………………….
Babak Farshchian

…………………………………………………………………….
María Óskarsdóttir

Date

Bjarki Freyr Sveinbjarnarson
Master of Science

Acknowledgments

I would like to thank my supervisors Anna Sigríður Islind and Erna Sif Arnardóttir, for their supervision throughout the project. I also want to express my gratitude to the rest of the Sleep Revolution Project members for their advice and recommendations, their help, and for giving me the privilege of working on such an exciting project. Furthermore, I would like to thank the people who participated in the usability and user experience tests. Finally, I would like to thank my family for supporting me spiritually and financially throughout my education. This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 965417.

Contents

1. Introduction **1**

2. Background **3**
- 2.1 Data 3
- 2.2 Sleep Measurements 4
- 2.3 Software Architecture 4
- 2.4 Relational Databases 6

3. Methods **8**
- 3.1 Research Context 8
- 3.2 Selection of Architecture 9
- 3.3 Requirements and Specification 9
- 3.4 Experiment Design 11

4. Implementation and Evaluations **13**
- 4.1 Layered Architecture 13
- 4.2 Use Case Extraction from Noxturnal 15
- 4.3 Display, Usability and User Experience 15
- 4.4 Performance Evaluation 20

5. Discussion and Conclusion **22**

Appendix **27**
- Images 27

1. Introduction

It has become safe to say that the modern world is information-driven, as we collect an increasing variety of data types and larger volumes across society. That is visible through the exponential rate at which we are gathering data. According to Statista [1], from 2010 to 2020, the total volume of created, captured, copied, or consumed data increased from around 2 to 59 zettabytes (59 trillion gigabytes). Furthermore, Statista expects the exponential growth rate to continue in the coming years [1]. Similarly, research is becoming more data-driven since some research fields require large amounts of various types of data [2]–[4]. Furthermore, new discoveries are made possible through increased variety and velocity in data [2]–[4]. Sleep studies are an excellent example of this since various sensors are often used simultaneously [4], [5]. Moreover, most physiological parameters are affected by sleep. Therefore the data used for sleep research has different origins, collected day or night, including physiological signals, cognitive testing, etc. [5]. Furthermore, the data used for sleep research purposes can be of a subjective and objective character [4].

When data is collected, like raw data, one can expect the data is not in a usable or analyzable format[6]. Since it is unusable, the next step is to turn the collected data into a structured form that can be used [6]. Therefore the usability of the data is low before it is processed and made into information [7]. There is a difference between "data" and "information" [7] since information is data that is processed to be beneficial [7]. Information, for example, often answers the questions of "who," "what," "where," and "when" [7].

In the Sleep Revolution project, a large EU-funded project operated at Reykjavik University in collaboration with 37 partners across Europe, numerous data types concerning sleep are collected for researchers to test their hypothesis on over 10,000 sleep measurements [8]. The project also uses other data sources such as questionnaires, various sleep-measuring devices, smartwatches, and neuropsychological tests. However, a view or filter is essential to use and understand the gathered data. Furthermore, the data types, extraction methods, and the usability of such a system will most likely be changing throughout the Sleep Revolution project, making adaptability for the extracting system necessary.

The primary aim of this thesis was to design and develop a system that can extract metadata from multiple sleep measurements to make it easier for end-users to understand what the data contains. For this to work effectively and due to the different origin and characteristics of the available data, the end-users need to be able to view the data in a way that allows them to identify the information the data contains, like statistical power, correlation, and information coverage. The thesis goes into the design of a system that will enable extracting various data into a standardized structure, which demonstrates the information of the data and allows the data with specific characteristics to be isolated and visualized effectively. A highly adaptive architecture is required since the system must serve data exploration for future research hypotheses. Furthermore, an adaptive architecture makes system expansion and integration easier for future designs. Using layered architecture is common to design and develop systems in a more adaptable way, but this raises the questions: 1) *How can we develop a system to extract data from scored sleep measurements?* 2) *How can layered architecture be used to read, display and filter the extracted data?* In the beginning, the thesis will look into data, data structures, sleep measurements, software architectures, and relational databases. Then the thesis shows the set requirements and the selection of both the database and the architecture. Finally, the thesis goes over the results and how layered architecture makes the system more adaptable.

2. Background

2.1 Data

Gathering data has always been a mandatory part of research since the fundamental role of every research project rests in data. However, researchers often use more than one data collection method resulting in multiple data types [9]. Systematic data collection methods include; i) surveys, ii) interviews, iii) focus groups, iv) observation, v) data extraction from sensor data, and vi) secondary data sources [9]. Furthermore, when research requires large amounts of subjective and objective data, such as research using polysomnography (objective data) and sleep diaries (subjective data), it can prove challenging to understand the information within the data [4]. Similarly, it can become difficult for people to understand the information if the data has many entries or if the data format is complex. Furthermore, if the data originates from different sources, the data can have multiple formats. In such cases, metadata modularity can be the key to organizing and understanding the information within the data [10]. Therefore, in a system representing diverse or new data types, adaptability, extensibility, and scalability must be accommodated [10].

When creating metadata, two valuable refinements make the metadata better [10]. The first refinement is combining multiple elements that represent the same thing. For example, different data sources might use “test subject,” “subject,” and “name,” which all represent the same subject’s name. These elements can all be combined, like using “name” for all of them. Another refinement is defining the range or the format of a data type. For example, the encoding of time can be represented as “yyyy-mm-dd,” “dd/mm/yy,” or “July 5th, 2002 15:30”. These formats can all be defined in the same format, like epoch time, which is total seconds from January 1, 1970. Furthermore, some types of metadata, like name order or dates, can be read differently in different countries; for example, 01/02/2020 represents February 1st and January 2nd. Therefore often, the ambiguity or the issue of multiculturalism needs to be addressed [9]. Combining the different formats into a single element makes the usability of the system better [10]. Therefore, the method of associating the metadata with the collected data is crucial since it might not be harvestable depending on the association model [10].

Data structures are frequently used to describe a collection of data. Data structures can be of three types: i) unstructured, ii) semi-structured, or iii) (fully) structured. Unstructured data is often data that cannot be described in a relational database [11]. Examples of unstructured data are texts, images, and videos. The main disadvantage of unstructured data is that controlled navigation within the data is impossible [11]. Therefore, using unstructured data can become increasingly difficult when a lot of data is collected. Semi-structured data has markers such as tags to separate elements but does not follow the structure of data models of data tables or relational databases [12]. One of the more significant advantages of semi-structured data is that it is unconstrained [12]. Furthermore, using schemas, the semi-structured data can be converted into (fully) structured data [12]. (Fully) structured data is when the data organization is predefined before it is populated, like in relational databases [11]. The performance and navigation are higher in (fully) structured data than in semi-structured and unstructured data, but they also have less flexibility and a lower level of scalability [11].

2.2 Sleep Measurements

Everybody sleeps, and on average, people spend a third of their lives sleeping. The importance of sleep has been increasingly emphasized throughout the years [3]. For example, the record for the longest an individual has gone without sleep is about 11 days [13], shorter than one can go without food. As we understand the importance of sleep better, we know more about sleep disorders and their effects on psychological and physiological health. To understand how the quality of sleep is measured, we need to understand its characteristics. There are multiple sleep disorders, and people with sleep disorders are often unaware of having them [4], [14]. For example, obstructive sleep apnea patients are commonly unaware of frequently waking up for short periods and unaware of their breathing stopping (apnea) [4]. The severity of sleep apnea is split into four stages, depending on the average AHI (Apnea-Hypopnea Index) during sleep; i) normal (AHI < 5), ii) mild sleep apnea (AHI is between 5 and 15), iii) moderate sleep apnea (AHI is between 15 and 30), and iv) severe sleep apnea (AHI > 30) [15]. A sleep study or polysomnography uses multiple sensors simultaneously to monitor the brain activity, heart rate, respiration, blood oxygen level, and various other signals, which indicate what is happening during the subject's sleep [3], [4], [14]. Those signals are recorded throughout the night from before the subject goes to sleep until the subject wakes up [16]. Then, the collected signals are used to understand the events and sleep stages throughout the night to detect sleep disorders and their severity [14], [16]. Processing and annotation of a sleep study are known as "scoring" a sleep study [4]. The scoring can be done by a sleep expert or automatically [4].

A polysomnography study uses an electroencephalogram, which monitors the brain waves, and an electrooculography that measures eye movements. These measurement tools monitor non-rapid eye movement, which splits into three sleep stages (N1, N2, N3), and the last sleep stage; rapid eye movement sleep [4], [14], [16]. Let's take a look at obstructive sleep apnea as an example. Obstructive sleep apnea is a disorder where the subject's breathing repeatedly stops throughout the night [3], [14]. Some of the nocturnal symptoms observed during obstructive sleep apnea are loud snoring, cessation of breathing, and respiratory effort [14]. Many things happen during an obstructive breathing event. For example, the patient struggling to breathe results in increased muscular activity, decreased oxygen saturation, and irregular respiratory signals [14], [17]. Furthermore, the body behaves differently in each sleep stage, making it more likely to get events in some sleep stages than others [17]. Objective data, like data collected with the sensors, is not the only data used to detect sleep disorders [4]. Subjective data is also used to screen for obstructive sleep apnea symptoms collected using, for example, questionnaires, interviews, or sleep diaries [4]. The subjective data is an essential tool for diagnosing and treating sleep disorders like insomnia or restless legs syndrome [4].

2.3 Software Architecture

It is essential to have some kind of arrangement of all of the components in software. Albert Einstein said: "Any intelligent fool can make things bigger, more complex, and more violent. It takes a touch of genius - and a lot of courage to move in the opposite direction" [18]. Common reasons for using some form of architecture or structure to the code are to i) justify implementations or decisions, ii) having a better overview and more evident purpose, iii) reduce the complexity of the system, iv) making the code more reusable and less fragile to change and, v) depending on the chosen structure, the system can be made more scalable, flexible or have a better performance [19], [20]. Therefore, using an architecture that fits the system's design may be essential to fulfill the requirements.

Modular architecture is an approach to create independent modules that work as a part of a more extensive system [21]. Using modular architecture is common in software design and development to reduce the complexity of a system while increasing its flexibility [21]. Furthermore, since it is relatively easy to replace, update or remove created components in a modular architecture, it is considered somewhat adaptable [20], [21].

Layered architecture is an architecture that divides the code modules into layers or packages [19]. Each layer has a role within the system where the layer is a collection of components [19]. The layered architecture typically has three layers; Presentation/Contents Layer, Business/Service Layer, and Database/Device Layer [19], [20]. Layered architecture is also known as n-tier architecture, where n is the number of layers, and the role of each layer varies between systems [19]. For example, the presentation layer is responsible for making the system interactable with the user, communicating the requests from the user into the system [19]. The business layer takes the requests from the presentation layer and translates them into an actual logic, typically communicating with the database layer in the process [19]. The final layer, the database layer, is where the queries and access to the data occur [19]. Suppose the layers above can communicate with layers below an example layer. In that case, the example layer is considered *open*. However, if the layers above cannot directly communicate with the layers below the example layer, the example layer is *closed* [19]. Usually, designers try to keep all layers closed to reduce the number of dependencies in the system [19]. An example of a layered architecture is shown in Figure 2.1.

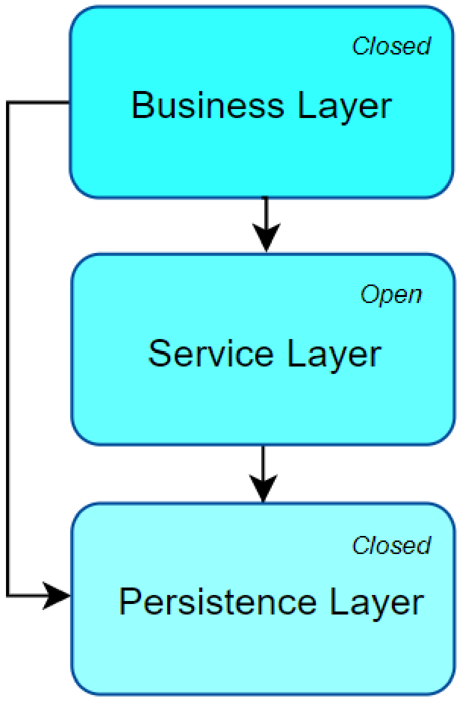


Figure 2.1: A layered architecture, where each layer has a different role. Layer L is considered closed if the layers above L can not communicate with the layers below L, but is considered open otherwise.

Another well-known architecture is Microkernel Architecture that has a core application

[19]. Plug-in components are additional features or components that extend the core system's capabilities [19]. The focus of the design for the core system is making minimal changes required to make the system work with the added plugins [19].

2.4 Relational Databases

The six most commonly used database management systems (DBMS) are reviewed in the following text [20]. DB-Engines is a website with a knowledge base of relational and NoSQL (not only structured query language) database management systems [22]. DB-Engines was used to get the most popular DBMS. Although DB-Engines also reviews NoSQL DBMS, the following text only includes a review of relational DBMS due to the highly relational nature of the topic at hand. The top six DBMS are Oracle, MySQL, Microsoft SQL Server, PostgreSQL, IBM Db2, and SQLite [22].

Oracle is the highest-ranked DBMS by DB-Engines, due to the number of mentions of their system, general interest, number of professional networks, etc. [22]. Oracle DBMS was made to reliably manage large amounts of data where many users can simultaneously access the same data [23]. The simultaneous access is made possible using a client/server architecture, where the Oracle DBMS uses distributed processing to handle multiple requests [23]. Furthermore, Oracle is also considered to have the most advanced database technology [24]. However, Oracle is not free to use besides personal use [23].

The second most popular DBMS is MySQL. According to MySQL, it is the most popular open-source DBMS [25]. Like Oracle, the MySQL DBMS is a high-performance, multi-user, and robust DBMS [23], [25]. Furthermore, since it is open-source, it is free to use under the GNU (GNU's Not Unix) General Public License terms but is also available for a standard commercial license [25].

Microsoft SQL, the third most popular DBMS [20], has multiple editions of SQL Server that are picked depending on the performance, runtime, and other requirements [26]. According to Microsoft SQL, it offers good scalability, performance, availability, and security [26]. However, unlike MySQL, it is not free to use except for development purposes [25], [26].

PostgreSQL, the fourth most popular DBMS[20], is an open-source object-relational database management system that supports a large part of the SQL standard [27]. It has a liberal license, making the users freely available to use, modify or distribute PostgreSQL [27]. The PostgreSQL community prides itself on being open-source since multiple developers have helped with identifying and correcting bugs and making improvements [27].

IBM Db2, the fifth most common DBMS [20], is an open-source relational database that also promises high performance, data availability, and reliability [28], [29]. According to IBM themselves, the IBM Db2 is "Powered by artificial intelligence and built to empower intelligent businesses with multi-modal management" [29]. They also have several tools that provide backups, recovery, security, and encryption [29].

The sixth most common and the last DBMS reviewed is SQLite. SQLite is used in smartphones, Mac and Windows operating systems, and various browsers, making SQLite DBMS likely to be the most used database engine in the world [30]. It is open-source and free to use for any purpose [30]. The codebase is worked full-time by an international team that continues enhancing the database [30].

A summary table of the six DBMS is shown in Table 2.1.

Database Management System	Free to use	High performance	Other
Oracle	No	Yes	The most advanced Database Management System technology.
MySQL	Yes	Yes	Large community, good documentation.
Microsoft SQL Server	No	Yes	Great security and backup.
PostgreSQL	Yes	Yes	Large community.
IBM DB2	No	Yes	Great for artificial intelligence related products.
SQLite	Yes	Yes	Most used Database Management System in the world.

Table 2.1: Shows summary of the six Database Management Systems discussed.

3. Methods

Action Design Research (ADR) is a research method used to evaluate a digital product [31]. ADR is split into four stages: i) Problem Formulation, ii) Building, Intervention, and Evaluation, iii) Reflection and Learning, and iv) Formalization of Learning [31]. The first stage, Problem Formulation, includes creating a design, formulating research questions, defining roles, and more that can be evaluated [31]. In the second stage, the design is implemented, intervened, and evaluated. The third stage is continuous, performed parallel with the first two stages, where the work is continuously reflected upon, and observations are noted [31]. Finally, in the last stage, the learnings of the design are formalized [31]. The overall method applied in this thesis is ADR since continuous learning outcomes were noted and shaped the design further, which is similar to the third ADR stage [31]. Furthermore, the design was made with evaluation in mind, a pivotal point in the first stage of the ADR [31]. Moreover, the usability test and evaluations were designed to contribute to the system's future designs, a critical part of the second stage of the ADR [31]. Finally, design principles were made that might contribute to designs of similar problems, which is precisely what the fourth stage in the ADR, Formalization of Learning, is about [31]. However, the research methods used in this thesis do not include all of the tasks and principles in ADR [31]. For example, the task to discover the initial knowledge-creation target does not appear in this research method, nor does the principle to create mutually influential roles [31].

3.1 Research Context

In the Sleep Revolution project, the goal is to collect a large set of sleep data allowing researchers to mine the data for answering fundamental questions on sleep, sleep disorders, and how it may be alternatively measured [8], [32]. The collection and processing of the sleep data are the core subject of the project. Therefore, to deliver reliable results, the following is of utmost importance; i) the possibility of choosing parameters to describe the characteristics of the data, ii) understanding the coverage of the data for different subjects, iii) the importance of that coverage for each subject characteristics [32]. The typical parameters used to describe sleep data include; i) age, body mass index, gender, weight, and height, ii) obstructive sleep apnea severity, or AHI, iii) index for periodic limb movements during sleep, iv) sleep profile in different sleep stages and total sleep time [3], [14]. In addition, Sleep Revolution will add more statistical parameters to process from the raw sleep data. Furthermore, all combined parameters for a sleep study can be looked at as the study's characteristics fingerprint [32].

Based on those parameters, the idea behind this project is to fetch or pull from the collected data research, training, and testing datasets for exploratory conclusions. For efficiency and to avoid unsuccessful use cases, it is vital to examine the data. Furthermore, it is necessary to understand what information the data contains since it may be too shallow for statistical significance. Additionally, it is made possible to see what parameters are dependent and what parameters are independent. For example, the coverage of all periodic limb movements during sleep and obstructive sleep apnea patients might not be enough for research on patients with both of them. Understanding the importance of the parameters describing the data could then potentially create the most informatic training and testing datasets possible. Additionally, to create a measure of coverage, allowing data that was not covered by the training and test datasets to be identified and directed to human-in-the-loop processing in real-world applications. Therefore, when the raw data is reprocessed for mining additional parameters, it will result in other data types in the database. However, the purpose of the database is not to answer research questions but rather to locate relevant sleep studies of interest for reprocessing to answer the research questions. Therefore, criteria for additional columns to the database should then represent the searchability in the database rather than answering all possible future questions.

3.2 Selection of Architecture

In the created extracted database management system (EDBMS), a database was necessary to store the extracted information. A decision was made to use relational DBMS since non-relational databases are primarily suitable for large amounts of unstructured data [24], which is not the case for this system. The selected database needs to store more than 10,000 sleep studies and be searchable or filterable [8]. As mentioned above, the choice included the six different DBMS. Since the DBMS has to be free to use for commercial purposes, the Oracle and Microsoft SQL Server could not be used [23], [26]. The IBM Db2 is heavily focused on artificial intelligence [28], and based on that; it was not deemed a good fit for this thesis. SQLite underperforms PostgreSQL and MySQL regarding multi-user capabilities, user access management, and security [33]. MySQL is considered more compatible with other systems than PostgreSQL [24], [33] and PostgreSQL has less documentation support [24]. Based on this, a decision was made to go with MySQL over the other DBMS.

In this thesis, where EDBMS for extracting information from data was designed and developed, a software architecture was used. The software architectures considered were; i) Layered Architecture, ii) Modular Architecture, and iii) Microkernel architecture. The first software architecture, layered architecture, was a fitting choice for several reasons. First of all, the EDBMS needs to fit other applications or systems with minimal changes. For example, the EDBMS displays, reads, and filters extracted information on a website. However, it needs to be reusable since it might be used for other platforms such as desktop applications or phone applications. Changing platforms requires changes to the presentation layer in a layered architecture, but not to the rest of the layers. The EDBMS also has functionalities that might be of use for other systems. Therefore, direct access to one of the layers can be the only necessary change for the new system to use the EDBMS, making the system used similarly as a library. The second architecture, Microkernel Architecture, was found not a good fit for the EDBMS since the design of the EDBMS is not based on plugins. Since Microkernel Architecture is made of core-system and plug-ins [19], it is hard for other new systems to adapt the functionalities of the EDBMS to their own. Modular Architecture might have been a good fit for the EDBMS. However, the layered architecture uses a modular architecture and has a more fitting structure [20]. Even if both Microkernel and Modular Architecture can outperform the layered architecture in performance or scalability, neither have an agile design [19].

3.3 Requirements and Specification

The EDBMS can extract information from already processed or scored sleep studies. Therefore, the EDBMS will not contain any raw data but only structured information extracted from scored sleep measurements. The EDBMS also assumes it receives only measurements that are usable for the end-users. The assumed preliminary process for the sleep studies is shown in Figure S1 in the Appendix. The extracted information is, for example, information about the subject (gender, age, etc.), the measurement tools used (Nasal Cannula, electroencephalogram, audio recording), and the values and indexes scored (AHI, total sleep time, etc.).

The EDBMS extracts information from a software called Noxturnal (NOXTURNAL®, Nox Medical, Reykjavík, Iceland), which is widely used to score sleep studies within the Sleep Revolution project. However, the EDBMS needs to be designed to use other data sources such as data gathered by smartwatches, sleep diaries, and other software. Moreover, each extraction process is most likely different for every data source since the software or documents store their data differently. Therefore it is essential to make the extraction process an independent component in the EDBMS. However, since the EDBMS is made with multiple extracting components in mind, format validation is essential. The format validator will ensure that the extracted data is in an acceptable format for the database in the EDBMS. The connection between i) data sources, ii) extractors and iii) the format validator is shown in Figure 3.1.

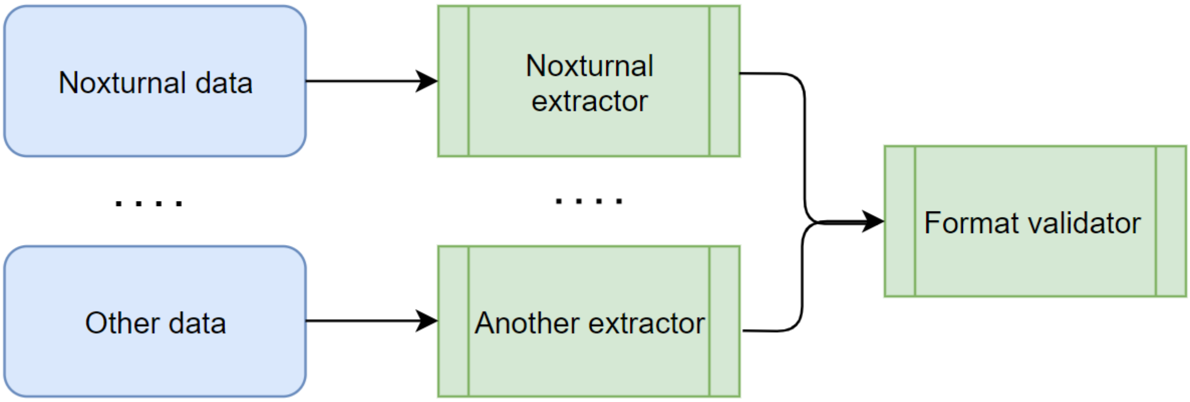


Figure 3.1: The figure shows how the process of data being extracted and then validated. A new extracting component is made for every new source of data.

The system gathers extracted data from different sources into a single database. A relational database is used so the stored and accessed data can be filtered and displayed. The users can see any data they have access to and set some filters through an interface. The filters are for the extracted values like age, dates, or AHI, which can be filtered using input boxes and dropdowns. Even if this system only displays the filtered data, it needs to be somewhat compatible with other systems, making it adaptable to newly created systems with minimal work. An example display is shown in Figure 3.2.

Filters:	ANY	40 - 55	20 - 30		ANY	ANY	ANY
Result Table							
Observation ID	Gender	Age	Body Mass Index	Date	Apnea–Hypopnea Index	Min Pulse	Max Pulse
HIDDEN	Male	40	25.0	20.02.2019	13.5	56	92
HIDDEN	Male	45	27.8	05.08.2019	16.8	62	108
HIDDEN	Female	54	22.0	05.01.2020	2.8	48	68

Figure 3.2: An example display with the result table. Above the table, the user can either

select or type to filter the results, based on the relevant values/indices.

The Sleep Revolution project expects a minimum of 10,000 sleep studies [8], which indicates that the EDBMS needs to store at least extracted data from 100,000 studies. Of course, the number of columns depends on the number of data sources extracted from the sleep measurement and the number of selected data types for the user. However, it is fair to say that one can expect at least 50 different relational data types in use at a time. Therefore, the system needs to hold at least 100,000 rows with 50 columns of data. Furthermore, the number of columns in the database must be extendable since hundreds of data types will be gathered later on.

3.4 Experiment Design

Every system needs a performance evaluation to understand its limitations. Since the most time-consuming functionality of the system is filtering the data, the objective of the tests is to see how long it takes to filter an increasing amount of data. It is expected that the MySQL database can handle the maximum number of filters of the maximum amount of expected data within a few seconds [34]. Furthermore, the scalability of the database is essential; therefore, it is necessary to see how the database can filter an increasing amount of data. Thus the tests can estimate the limitations of the system after the scalability has been evaluated.

The EDBMS is required to handle at least 100,000 measurements. For this thesis, the tests and the created system will use simulated data for display. When the information is extracted from Noxturnal, most of it will be in string format (names, ID, devices) or number format (indexes, dates, values, amounts). Therefore, the performance test will evaluate i) a database consisting of strings or varchar, and ii) a database consisting of numbers or floats. The starting size of the database will be 100 entries. The size of the database will then continually increment by 100 until 100,000 entries have been reached. The performance test will use a filter on each of the 50 columns every time the database is extended.

A usability and user experience test was performed to evaluate the usability of the system. Ten experts took the usability test. The test contained a filterable database with 100 simulated results, where the results are similar to data from actual sleep measurements. The results were displayed as a table and interactable filters, as shown in Figure 3.2 above. The test subjects were given a short introduction to the system before the test started. Then, the prototype testers were given tasks to use the filter and database to answer four predefined questions:

i) How many subjects in the database are male and how many are female?
ii) Is the AHI more severe (higher) for aged 40+ or less than 40?
iii) How many males in the database have severe (30 or higher) AHI?
iv) What is the average weight for males, and what is the average weight for females with moderate apnea (15 to 30) AHI?

The answers were in the format of text inside Google Forms. Google Forms also contained their responses. Each of the tasks was timed to evaluate their difficulty.

The evaluation uses two other evaluation formats, the first one being AttrakDiff [35], consisting of 25 questions where the test takers evaluated the user experience of the database and the filters they used for the first task. The data from the ten participants were combined into a matrix and analyzed to understand the user experience on the one hand and usability on the other hand. The questions were in a format where the prototype testers gave ratings between two antonyms, as shown in Figure 3.3.

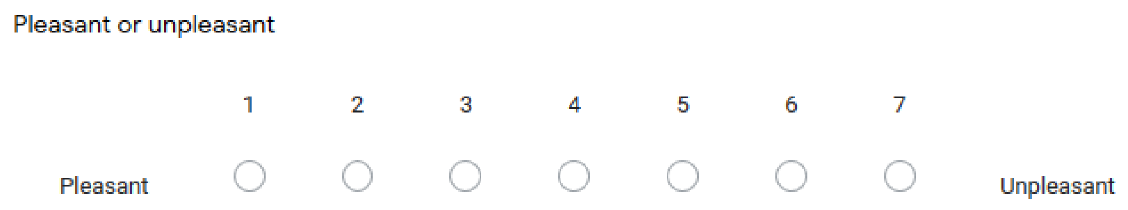


Figure 3.3: An example question in the AttrakDiff evaluation.

The second evaluation format is the System Usability Scale [36], consisting of 10 questions, focusing more on the database's usability. An example question of the System Usability Scale evaluation is shown in Figure 3.4.

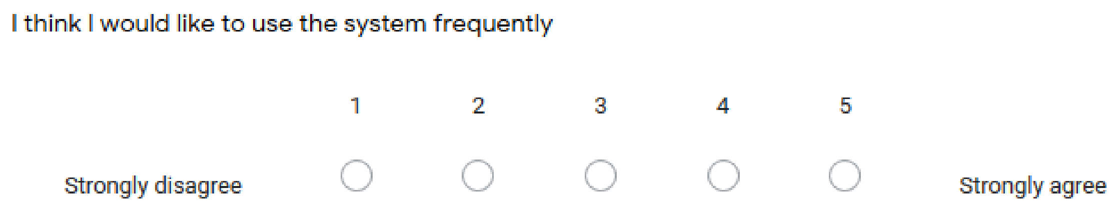


Figure 3.4: An example System Usability Scale question.

Finally, a semi-structured interview was conducted, which took between 2 to 20 minutes. The time for each person reflected heavily upon improvements that might benefit their work within the Sleep Revolution, which was the central theme of the interviews. Therefore, the results were about the user experience. The usability experience was evaluated with quantitative measures with the System Usability Scale and AttrakDiff questionnaires. On the other hand, qualitative data from the semi-structured interview with experts based on their experience from testing the system.

4. Implementation and Evaluations

4.1 Layered Architecture

The system consisted of extracting information from different sources, storing the information in a database, and displaying the resulting database in an intractable way. Furthermore, a layered architecture was designed for the system initially to ensure the system's adaptability. However, two layered architectures were necessary since the system consisted of two independent subsystems; one for extracting data and the other for filter and display. The layered architectures are shown below in Figure 4.1.

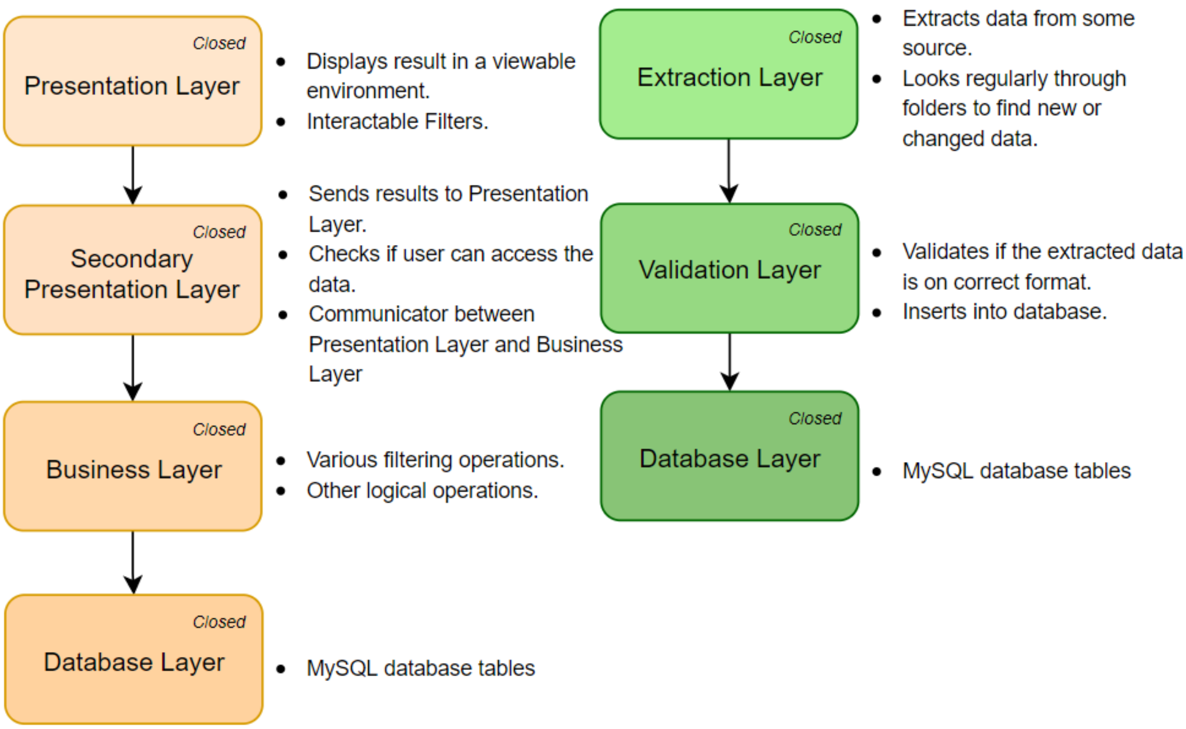


Figure 4.1: The layered architecture for filtering and displaying the information is shown on the left and extracting and inserting data on the right. The two subsystems are independent of one another, but both access the same MySQL database.

The Presentation Layer is where the view or display is made. Multiple applications can have different presentation layers but access the same Secondary Presentation Layer. The Secondary Presentation Layer is a collection of functions that the various components in the Presentation Layer use. For example, the Secondary Presentation Layer has a function to convert some filtered results into a JSON (JavaScript Object Notation) structure. This JSON structure can be preferred by a web application or a phone application. Therefore, when a new function is made, it depends on its reusability if it belongs in the Presentation Layer or the Secondary Presentation Layer. Finally, the Business Layer is the collection of logical functions. The reusability of the Business Layer is vital since it is likely new software or systems would communicate directly with the Business Layer instead of the Secondary Presentation Layer. Therefore another layered architecture is used for the Business Layer. The four sub-layers of the Business Layer and their purpose is shown in Figure 4.2.

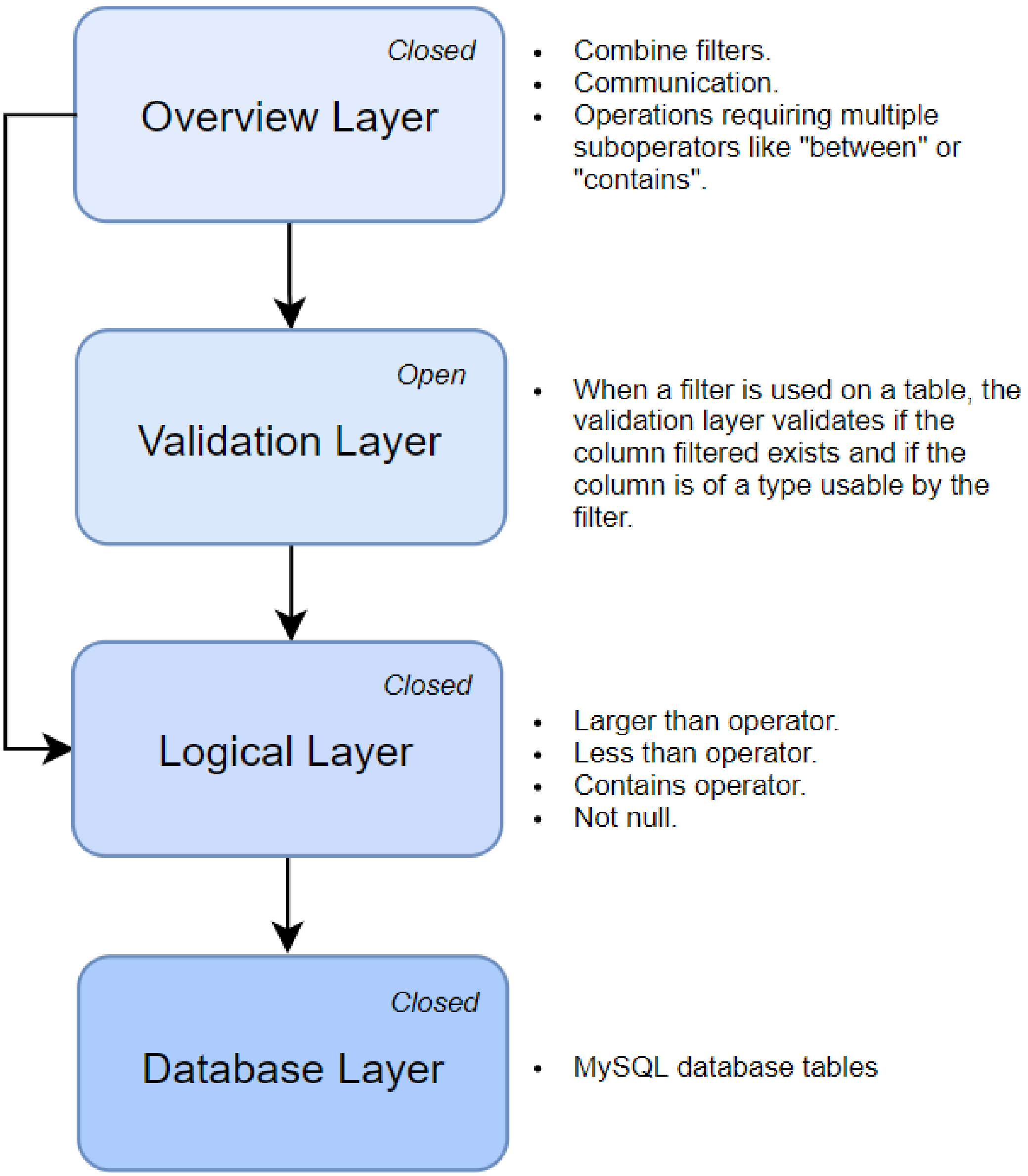


Figure 4.2: The architecture of the Business Layer, with four sub-layers, containing an open Validation Layer between the Overview Layer and the Logical Layer.

4.2 Use Case Extraction from Noxturnal

Before implementing the automated extraction process from Noxturnal, access to sleep studies measurements was necessary. After exploring different options for approaching the Noxturnal data, it was decided to extract the data directly from one of the output files. As the Noxturnal data files are proprietary and not published, some exploration work was needed to find a way to access the data. This work demonstrated that one of the proprietary files of "ndb" format was based on SQLite3 structure and could be read as such. Furthermore, the parameters in the database were found to have descriptive labels referring to the parameters reported by Noxturnal. This allowed the parameters to be extracted directly from the “ndb” file and processed to fit the selected format of the EDBMS. It should be noted that even if Noxturnal file format did prove to be readable, it could be expected that other vendors use less standardized structures or proprietary binary formats that can not be adapted as quickly as can be done with Noxturnal without the direct involvement of the vendor. The chosen variables for extraction were indexes frequently used to describe sleep disorders like AHI and values representing the patient like name, gender, body mass index, length of recording, and the device info.

4.3 Display, Usability and User Experience

After the layered architecture was designed, the filtering and display were implemented using Python and HTML. The system used the flask library for the get and post requests required for filtering. In addition, the system used a MySQL connector library to extract and filter the data. The primary purpose of designing a display was to evaluate the usability of the system better. The designed website offered an interactable filter with inputs for some of the columns. The filter made it possible for users to search the database depending on gender, obstructive sleep apnea severity, body mass index, etc. The filter is shown in Figure S2 in the Appendix. In addition, the user can view the average of all numeric columns below the filter and the filtered database. An example of the filter results is shown in Figure S3 in the Appendix. As stated above in the Experiment Design chapter, a usability test consisting of three parts was evaluated on ten prototype testers. The prototype testers consisted of six members of the Sleep Revolution project, two sleep experts, and two University students with a background in research and databases. The first part of the usability test consisted of four questions where the testers used the database and filters to answer. The results of the first part of the questionnaire are shown in the following table below.

Question	Answered correctly	Managed to complete task
How many subjects in the database are male and how many are female?	100%	100%
Is the AHI more severe (higher) for aged 40+ or less than 40?	100%	100%
How many males in the database have severe AHI (30 or higher)	90%	100%
What is the average weight for males and what is the average weight for females with moderate apnea (15 to 30) AHI?	100%	100%

Table 4.1: The questions asked, correctly answered and completion rate for the first part of the usability test where testers used the filter user interface to answer the questions.

Each question was answered correctly by all ten testers, except one, where the tester used Apnea Index instead of Apnea–Hypopnea Index. The average time taken was under 30 seconds for each of the questions except the last one, “What is the average weight for males and what is the average weight for females with moderate apnea (15 to 30) AHI?”, which was about 45 seconds. The tests were taken in person for 8 out of 10 participants. The remaining two took the test online; one took place on Discord and the other on Messenger, using the share screen functionality. However, as is shown in table 4.1, all of the testers managed to complete all tasks, and all questions except in one case were answered correctly. Therefore the tasks were most likely too easy to perform and answer, and the tasks should be made more challenging in future evaluations.

In the AttrakDiff questionnaire, many of the questions felt unrelated to the questionnaire. For example, the question “Is the system isolating or connective” made the participants unsure of what was being meant. As stated above, the participants could choose numbers from 1 to 7 where the number represented two contradicting words, as is shown in Figure 3.3. The following table shows the results for each question in the AttrakDiff questionnaire.

Question	Average	Standard Deviation
Human (1) or technical (7)	4.0	1.2
Isolating(1) or connective (7)	4.4	1.0
Pleasant (1) or unpleasant (7)	2.1	0.8
Inventive (1) or conventional (7)	4.3	1.8
Simple (1) or complicated (7)	1.6	0.7
Professional (1) or unprofessional (7)	1.7	0.9
Ugly (1) or attractive (7)	4.9	1.4
Practical (1) or impractical (7)	1.7	0.5
Likeable (1) or disagreeable (7)	1.6	0.7
Cumbersome (1) or straightforward (7)	6.3	0.6
Stylish (1) or tacky (7)	3.7	1.3
Predictable (1) or unpredictable (7)	1.9	1.2
Cheap (1) or premium (7)	4.5	1.0
Alienating (1) or integrating (7)	6.0	0.9
Brings me closer to people (1) or separates me from people (7)	3.5	1.0
Unpresentable (1) or presentable (7)	5.6	0.9
Rejecting (1) or inviting (7)	5.5	1.1
Unimaginative (1) or creative (7)	4.5	1.1
Good (1) or bad (7)	2.1	1.1
Innovative (1) or conservative (7)	2.1	1.6
Dull (1) or captivating (7)	4.5	1.3
Undemanding (1) or challenging (7)	2.4	1.0
Motivating (1) or discouraging (7)	3.2	1.2
Novel (1) or ordinary (7)	4.0	1.3
Unruly (1) or manageable (7)	6.4	0.7

Table 4.2: The results of all 25 AttrakDiff questions in the Usability Test, their average score, and their variance.

Testers found the database simple, professional, practical, likable, straightforward, and undemanding rather than their antonyms. However, the design needs to be improved since testers gave an average rating to questions regarding the system's design, style, or look.

The System Usability Scale questionnaire focuses more on how easy the system is to use. Participants in the usability test rated nine different questions from 1 (strongly disagree) to 5 (strongly agree). The results are shown in the following table.

Question	Average rating	Standard Deviation
I think I would like to use the system frequently	4.1	0.8
I found the system to be simple	4.7	0.5
I thought the system was easy to use	4.8	0.4
I think that I can use the system without the support of a technical person	4.9	0.3
I found the various function in the website were well integrated	4.2	0.6
I would imagine that most people would learn to use the system very quickly	4.5	0.5
I found the system very intuitive	4.1	0.7
I felt very confident using the system	4.3	0.6
I could use the system without having to learn anything new	4.6	0.5

Table 4.3: The results of the System Usability Scale questionnaire with each question and the average rating. The rating represents how much the participants agree with the question from one (strongly disagree) to five (strongly agree).

The results of the System Usability Scale show the system can become a helpful and straightforward tool. However, the system still needs some improvements since many of the testers did not give full ratings. Many of the reasons for not full ratings were explained by the testers in the following semi-structured interview.

After the usability test, a semi-structured interview was performed about how the tester thought the system might contribute to the Sleep Revolution project. Following are ten pointers made by multiple interviewees about the system itself or changes to it that would benefit their work within the Sleep Revolution project.

Main pointers made in the semi-structured interview
A system is needed to simplify the extensive database.
Some kind of a search is necessary to find data.
Getting the paths of all recordings meeting some kind of criteria is necessary for machine learning projects.
It is necessary to search the database for those who are scoring the sleep measurements.
More types of data are needed for those who are scoring the sleep measurements.
A separate system should be made for the people working in the Sleep Revolution, who have access to everything.
The focus of future design for the system should not be customer-based but rather as a helping tool for the Sleep Revolution employees.
Separate databases should be made for each type of measurement.
The system should only be accessible inside the Reykjavík University to ensure security.
More display methods are needed to understand the filtered results better.

Table 4.4: A list of main pointers collected in the semi-structured interview part of the usability test.

The results show that the designed system is simple enough to use for everyone. However, users felt some functionalities were missing that would have made their experience better. For example, saving the filter input between searches would have made the system easier to use. Furthermore, the semi-structured interviews showed that the testers felt like a filterable and searchable database is necessary for many of the projects within the Sleep Revolution. However, the semi-structured interview could have included an introduction or other material to help the interviewees develop ideas. Doing so might have resulted in a broader discussion, more topics, and different thoughts on improvements.

4.4 Performance Evaluation

It is necessary to assess the performance of the system to evaluate its scalability. Since the system is based around filters, a decision was made to test its capabilities for both floats and string entries. It has yet to be decided how the configurations on the server will be. However, the experiment configurations are shown in Table 4.5 below.

Item	Value
Programming language	Python
Development environment	PyCharm
Operating System	Linux
Libraries	MySQL connector, random
Motherboard	Z270X-Gaming K5
Processor	Intel(R) Core(™) i7-7700K CPU @ 4.20GHz, 4201 Mhz, 4 Cores, 8 Logical Processors
Graphics Card	NVIDIA GeForce RTX 2080 TI

Table 4.5: The configuration used in the performance evaluation tests.

The following figures, Figures 4.3 and 4.4, show the scalability of a database consisting of strings and floats. As can be seen in the statistics, the increase in time taken was linear. The increase in rows between iterations was 100 rows. Therefore a total of 1,000 tests were performed to get each result. The test used 50 columns of either strings or floats. The test used only strings and float data types since they are the most used data types in the Noxturnal database. It was decided to stop at 100,000 since the Sleep Revolution project expects only 10,000 sleep measurements [8]. Furthermore, since the results were linear, it is easy to evaluate the scalability of a more extensive database.

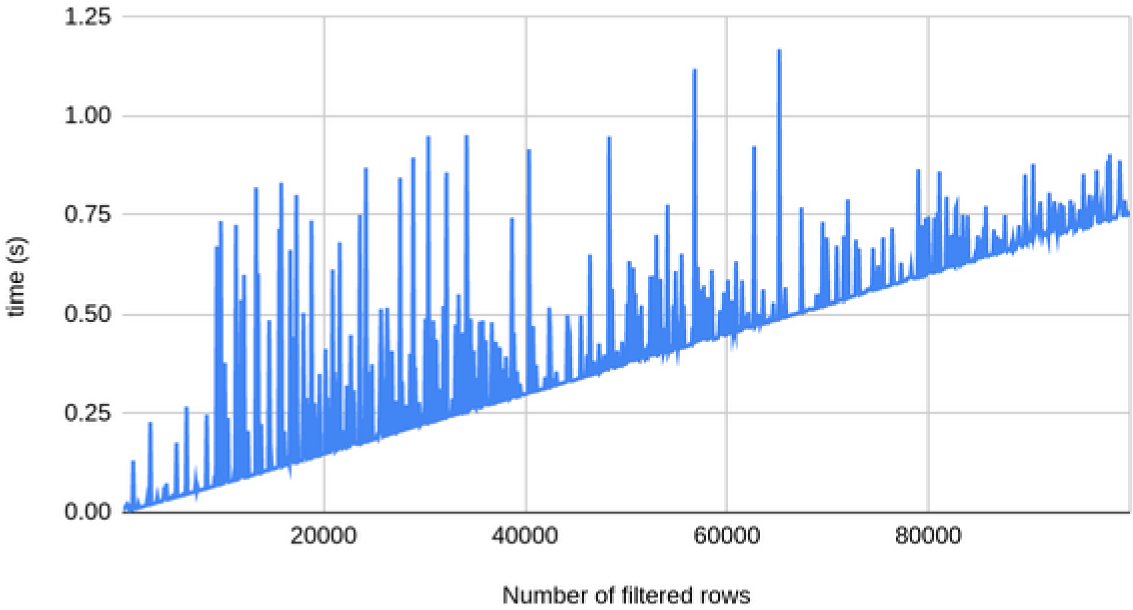


Figure 4.3: The time in seconds it took to filter strings in a database from 100 up to 100,000, increasing by 100 at a time, where each of the 50 columns was filtered. The filter was in the format “%z%z%” and the string inputs in the format “column_x_row_y” where x and y were the representative numbers of rows and columns in the table.

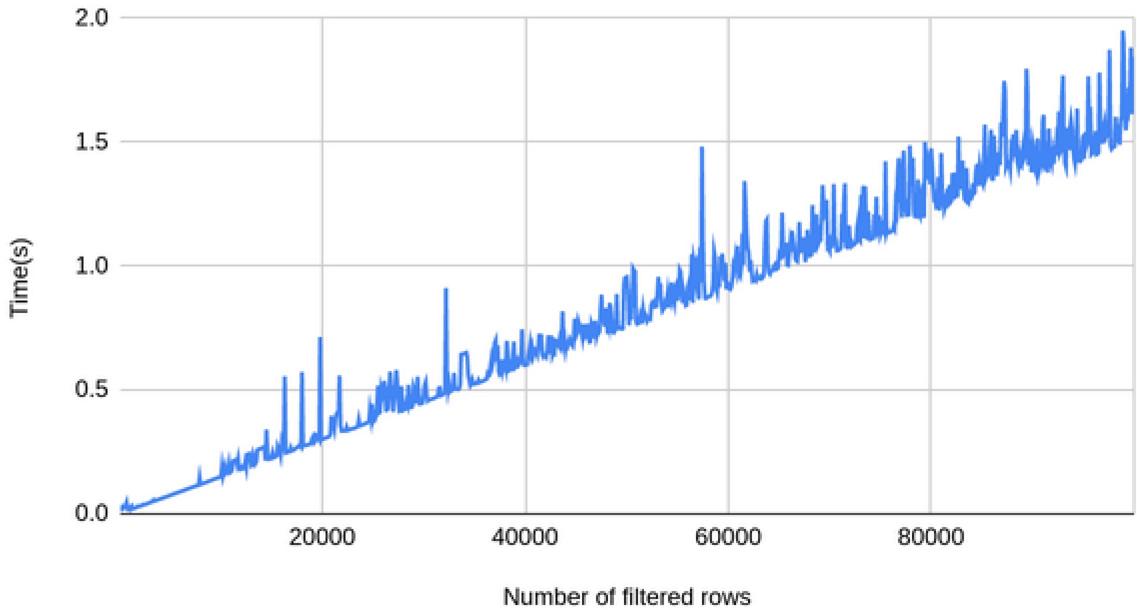


Figure 4.4: The time in seconds it took to filter floats in a database of size 100 rows up to 100,000 where each of 50 columns in the table was filtered. The inserted floats in the database were of the largest possible size (53). The floats were random floating numbers between 0 and 100. The filter used was “between 49.0 and 50.0”.

5. Discussion and Conclusion

As stated early on in this thesis, the modern world is information-driven and we collect an increasing variety of data types and larger volumes across society. However, there is a difference between “data” and “information” since information is data that is processed to be beneficial [7]. Information, for example, often answers the questions of “who,” “what,” “where,” and “when” [7]. Furthermore, there is a growing interest in data-driven research, since some research fields require large amounts of various types of data [2]–[4]. This is especially true for healthcare settings. In order to develop a system to extract data from scored sleep measurements, we need to understand how layered architecture can be used to read, display and filter the extracted data and that is essentially what this thesis is about.

In this thesis, an evaluation was made for an implementation of a designed system. The system consists of a filterable database and an extraction method to i) design and develop a system to extract data from scored sleep measurements, and ii) an layered architecture that can be used to read, display and filter the extracted data. The design and development of a system to extract data from scored sleep measurements were explored. There are different methods to store data. For example, the thesis used fully structured data instead of the other two data structures; semi-structured and unstructured. As stated in Chapter 2, the main advantage of using fully structured data is the performance but sacrifices flexibility instead [11]. However, since performance is of high importance, using fully structured data for filtering is necessary. Furthermore, unstructured data, such as raw sleep measurements, take a lot of storage, and navigation is impossible [11]. The layered architecture was chosen of the three considered software architectures. The primary reason for picking the layered architecture is how well the design fits the system and how the design makes the system more adaptable to changes and additions, as is shown in Chapter 4.1. In addition, the design uses validation layers where it is possible to make additional features easier to integrate and validate. The usability test and the performance evaluation show that the created system is expandable, useful, and scalable. For example, the usability and user experience evaluation shows that the system is simple to use since all testers completed each task. Furthermore, the semi-structured interview resulted in many ideas such as; i) additional types of data to extract, which could benefit their work, ii) more engaging display and interactions, such as boxplots or other statistical descriptives, iii) sortable columns, iv) comparable columns, etc. Moreover, fifty columns of either strings or floats were filtered with over 100,000 rows of simulated data in about 1.5 seconds in the performance evaluations.

The chosen methods, such as focusing on one 3rd party system for demonstrating direct extraction of parameters, were successful but simultaneously demonstrated the issue that may arise with other 3rd party systems that may be based on more proprietary data structures. Data structures that are not published may also be different between versions of the solutions and may even contain unpublished errors that the vendors deal with but could result in errors when extracted by exploration like was done here. It may indeed only be possible to reliably access the parameters from those 3rd party systems based on direct agreements with different vendors, direct input/output testing of interfacing with their black-box data systems, or getting the 3rd party vendors to provide special interfaces allowing extractions from their systems. Therefore, even if the one example of "black box" data extraction was successful through exploration and reverse engineering, this part will be different between vendors and fragile regarding versions and errors. In the Display, Usability, and User Experience evaluations, the semi-structured interview was the most contributing part of the evaluation for reasons including; i) the testers came up with multiple ideas, ii) the testers pinpointed what exactly they were unhappy with, iii) the discussion with each participant improved the talking points and focus of the discussion for the rest of the participants. The System Usability Scale evaluation was also useful since it confirms that users felt confident using the system, would frequently use it, etc. However, many of the prototype testers found the AttrakDiff questionnaire confusing. The AttrakDiff results also focused on similar things as the System Usability Scale evaluation resulting in little additional knowledge. As discussed in Chapter 2, there are different ways of gathering data to evaluate a product. This thesis uses observations, surveys, and interviews but does not use other data collection methods like a focus group. In the interview after the usability test, many discussions surfaced. Therefore, using a focus group would most likely have proven very useful where several people share their opinions [9]. Focus groups can enlighten other ideas, avoid repeated discussions, etc. However, in a focus group, there is less time for each individual to talk about his/her ideas, making it hard to find available time for everyone. Additionally, it may prove challenging to keep the discussion beneficial for the product.

As shown in the usability test, the system still needs more work before being a helpful tool in the Sleep Revolution project. Also, the system needs to be distributed to others through the Sleep Revolution cloud at Reykjavik University. In the usability test, people felt that some of the values in the Noxturnal were missing in the database. Therefore it is necessary to cooperate with the other Sleep Revolution researchers to decide what values should be extracted. Furthermore, to address the changing requirements of the extracted values, it is necessary to simplify modifications of extracted values. Therefore further research on simplifying the extraction method is essential. Additionally, vendor licensing and vendor interface agreement needs to be addressed as more systems are added. The database consists only of a single MySQL database table with all of the extracted information in the created system. Therefore, it is necessary to extract the data into different tables depending on the type of data. Doing so makes the system more scalable, may improve the system's performance, and reduce its complexity. When an end-user finds the data they want using filters; they may need to access it. Therefore, the display needs to offer some kind of an output method, for example, a text file that includes all of the locations of the filtered files. The system does not yet have a way to delete or modify inserted data. So, for example, if a scored measurement is changed, removed, or a new one is inserted, the system needs to detect it. However, before any changes are performed, the purpose of the system, what users it is for, where it is hosted, and more needs to be decided. An acceptance criterion needs to be made for adding a new parameter column to the database. The requirements will be decided on in collaboration with the Reykjavík University Sleep Revolution team.

Using MySQL was found to be a good choice for DBMS since the scalability is linear and MySQL is well documented. The performance test shows that filtering 50 different data types in over 100,000 measurements was done in under two seconds. Two seconds might be too long for some systems, but typically only a few filters are used for each search. The results of the usability tests show that significant work is necessary before the system can be used as a helpful tool for their work. However, due to the layered architecture design, expanding the system's capabilities is still a possibility. Also, since the prototype testers made several expansion proposals, it is necessary to evaluate each one, how important they are, and in which order they should be implemented. Therefore, a structured improvement should be formed and a prioritization group designated to rate the importance of each proposal. The designed layered architecture shows great promise for future work. The design was made adaptable for possible future changes, which was done in collaboration with

Reykjavík University Sleep Revolution members. The proposed changes by the prototype testers in the usability test have been prepared due to the early design of the system and the predicted future modifications and additions to the system. Therefore using layered architecture was found a fitting choice for the project. To elaborate, layered architecture is an architecture that divides the code modules into layers or packages [19]. Each layer has a role within the system where the layer is a collection of components [19]. From that notion, the system presented in this thesis is built and the findings show that using layered architecture was found a good choice for the project. However, the system still needs further work to become a helpful tool. Thus, the dependencies for success in creating a practical system depends on; i) if there is a lot of data in the database, ii) if all relevant data types are filtered, iii) if a server hosts the system with a reasonable response time, iv) if users can connect to the server, v) if the system has access to all of the data needed to be extracted, vi) if the data is researchable; that is, consent for the processing, ethics committee, and data security has been resolved, vii) if the system can update, remove and insert files automatically. The main contribution of the thesis can be combined into three design principles:

i) The system's design should be made of components with a single purpose. Such design makes it easier to replace, update, or even remove elements in the future.

ii) The use of validation layers makes additions to the system more straightforward, the results more credible, and the system more scalable.

iii) The use of usability tests and a semi-structured interview can be a valuable method to evaluate the product, pinpoint the users' problems, and get ideas for changes and additions.

References

[1] A. Holst, “Amount of information globally 2010-2024,” Feb. 2021.
[2] W. Raghupathi and V. Raghupathi, “Big data analytics in healthcare: promise and potential,” *Health Inf. Sci. Syst.*, vol. 2, no. 1, p. 3, Feb. 2014, doi: 10.1186/2047-2501-2-3.
[3] D. A. Pevernagie *et al.*, “On the rise and fall of the apnea−hypopnea index: A historical review and critical appraisal,” *J. Sleep Res.*, vol. 29, no. 4, p. e13066, 2020, doi: https://doi.org/10.1111/jsr.13066.
[4] E. S. Arnardottir, A. S. Islind, and M. Óskarsdóttir, “The future of sleep measurements – a review and perspective,” *Sleep Med. Clin.*, 2021.
[5] R. Mehra *et al.*, “Nocturnal Arrhythmias Across a Spectrum of Obstructive and Central Sleep-Disordered Breathing in Older Men: Outcomes of Sleep Disorders in Older Men (MrOS Sleep) Study,” *Arch. Intern. Med.*, vol. 169, no. 12, pp. 1147–1155, Jun. 2009, doi: 10.1001/archinternmed.2009.138.
[6] A. Labrinidis and H. V. Jagadish, “Challenges and opportunities with big data,” *Proc. VLDB Endow.*, vol. 5, no. 12, pp. 2032–2033, Aug. 2012, doi: 10.14778/2367502.2367572.
[7] G. Bellinger, D. Castro, and A. Mills, “Data, information, knowledge, and wisdom.” 2004.
[8] “Revolution of sleep diagnostics and personalized health care based on digital diagnostics and therapeutics with health data integration,” *Cordis*. https://cordis.europa.eu/project/id/965417 (accessed May 22, 2021).
[9] M. C. Harrell and M. A. Bradley, “Data Collection Methods. Semi-Structured Interviews and Focus Groups,” RAND National Defense Research INST Santa Monica CA, Jan. 2009. Accessed: Apr. 11, 2021. [Online]. Available: https://apps.dtic.mil/sti/citations/ADA512853
[10] E. Duval and W. Hodgins, “Metadata principles and practicalities,” *-Lib Mag.*, vol. 8, p. 2002, 2002.
[11] R. Sint, S. Schaffert, and S. Stroka, *Combining Unstructured, Fully Structured and Semi-Structured Information in Semantic Wikis*. European Semantic Web Conference (ESWC 2009), 2009.
[12] P. Buneman, “Semistructured data,” in *Proceedings of the sixteenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*, New York, NY, USA, May 1997, pp. 117–121. doi: 10.1145/263661.263675.
[13] S. Coren, “Sleep deprivation, psychosis and mental efficiency,” *Psychiatr. Times*, vol. 15.3, pp. 1–3, 1998.
[14] S. Chokroverty, “Overview of sleep & sleep disorders,” *Indian J. Med. Res.*, vol. 131, pp. 126–140, Feb. 2010.
[15] M. Friedman, R. Vidyasagar, D. Bliznikas, and N. Joseph, “Does Severity of Obstructive Sleep Apnea/Hypopnea Syndrome Predict Uvulopalatopharyngoplasty Outcome?,” *The Laryngoscope*, vol. 115, no. 12, pp. 2109–2113, 2005, doi: https://doi.org/10.1097/01.MLG.0000181505.11902.F7.
[16] J. V. Rundo and R. Downey, “Chapter 25 - Polysomnography,” in *Handbook of Clinical Neurology*, vol. 160, K. H. Levin and P. Chauvel, Eds. Elsevier, 2019, pp. 381–392. doi: 10.1016/B978-0-444-64032-1.00025-4.
[17] P. J. Strollo and R. M. Rogers, “Obstructive Sleep Apnea,” *N. Engl. J. Med.*, vol. 334, no.

2, pp. 99–104, Jan. 1996, doi: 10.1056/NEJM199601113340207.
[18] J. Fripp, M. Fripp, and D. Fripp, *Speaking of Science*. Newnes, 2000.
[19] M. Richards, *Software Architecture Patterns*. O'Reilly, 2015.
[20] Y. Yoo, O. Henfridsson, and K. Lyytinen, "Research Commentary—The New Organizing Logic of Digital Innovation: An Agenda for Information Systems Research," *Inf. Syst. Res.*, vol. 21, no. 4, pp. 724–735, Nov. 2010, doi: 10.1287/isre.1100.0322.
[21] H. A. Simon, *The Sciences of the Artificial*. MIT Press, 2019.
[22] "DB-Engines Ranking," *DB-Engines*. https://db-engines.com/en/ranking/relational+dbms (accessed Mar. 25, 2021).
[23] M. Cyran, P. Lane, and J. Polk, *Oracle Database Online Documentation*, 10.2. 2005.
[24] D. Tobin, "Which Modern Database Is Right for Your Use Case?," Jun. 2020. https://www.xplenty.com/blog/which-database/
[25] "MySQL 8.0 Reference Manual," *MySQL*. https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html (accessed Apr. 25, 2021).
[26] "Microsoft SQL Server 2019 Documentation," *Microsoft*, Mar. 2021. https://docs.microsoft.com/en-us/sql/sql-server/editions-and-components-of-sql-server-version-15 (accessed Apr. 25, 2021).
[27] "PostgreSQL 13.2 Documentation." Accessed: Apr. 25, 2021. [Online]. Available: https://www.postgresql.org/docs/current/
[28] "IBM Db2 – Data Management Software," Apr. 21, 2021. https://www.ibm.com/analytics/db2 (accessed Apr. 25, 2021).
[29] "IBM Db2: The AI Database," p. 10.
[30] "SQLite," *SQLite*. https://www.sqlite.org/about.html (accessed Apr. 25, 2021).
[31] M. K. Sein, O. Henfridsson, S. Purao, M. Rossi, and R. Lindgren, "Action Design Research," *MIS Q.*, vol. 35, no. 1, pp. 37–56, 2011, doi: 10.2307/23043488.
[32] "Sleep Science Friday: Time for a Sleep Revolution (#H2020) with Dr. Erna Sif Arnardóttir," *European Sleep Research Society*, Nov. 13, 2020. https://esrs.eu/ssf-2020-11-13/ (accessed Apr. 11, 2021).
[33] J. Hans, "SQLite vs MySQL vs PostgreSQL – The Search For The 'Best' Relational Database Management System," *Serverwise*, Jul. 2020. https://blog.ssdnodes.com/blog/sqlite-vs-mysql-vs-postgresql/ (accessed Apr. 25, 2021).
[34] B. Schwartz, P. Zaitsev, and V. Tkachenko, *High Performance MySQL: Optimization, Backups, and Replication*. O'Reilly Media, Inc., 2012.
[35] "AttrakDiff." http://www.attrakdiff.de/index-en.html#nutzen (accessed May 10, 2021).
[36] A. S. for P. Affairs, "System Usability Scale (SUS)," Sep. 06, 2013. system-usability-scale.html (accessed May 22, 2021).

Appendix

Images

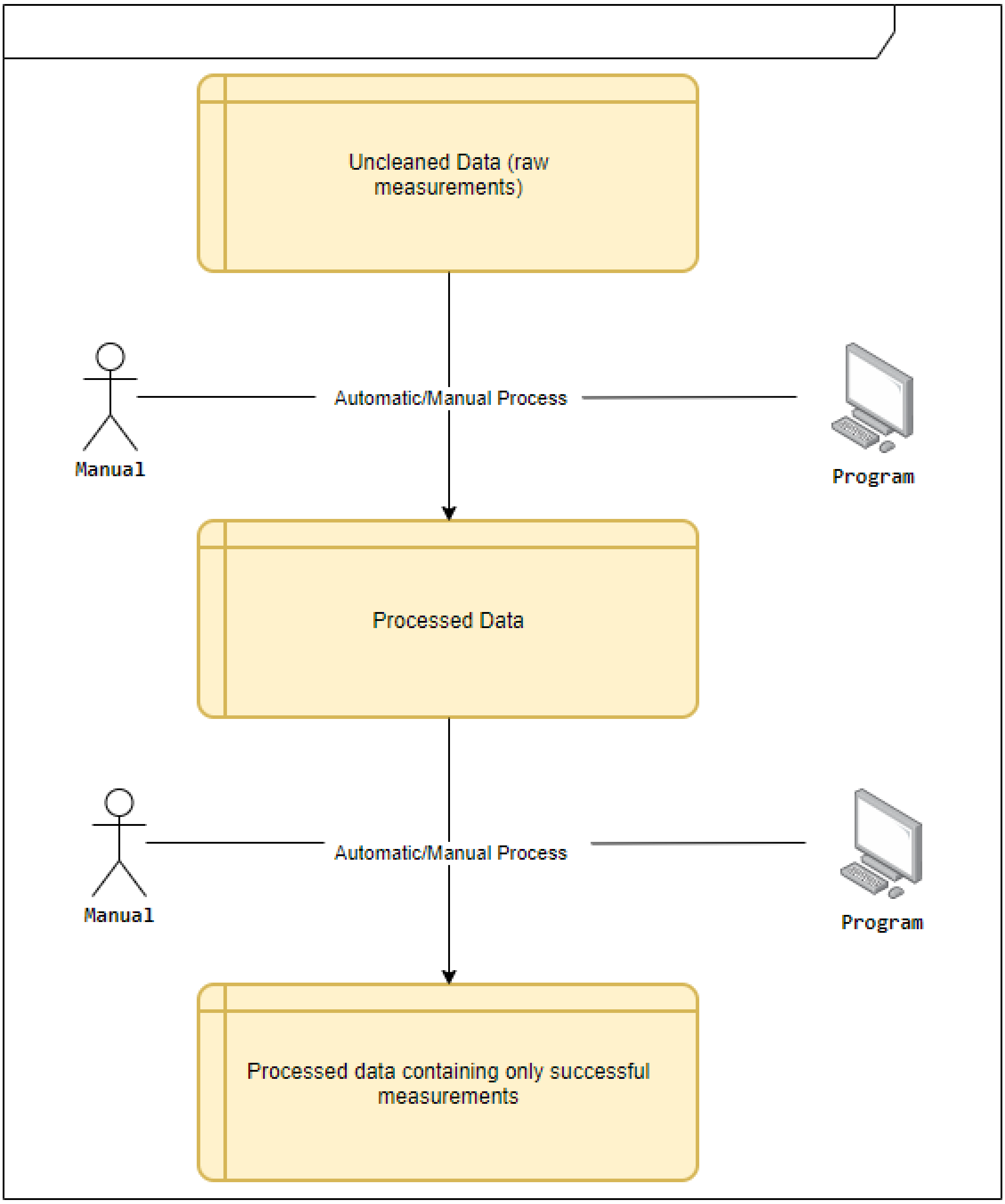


Figure S1: The assumed process of raw data to scored and usable measurements

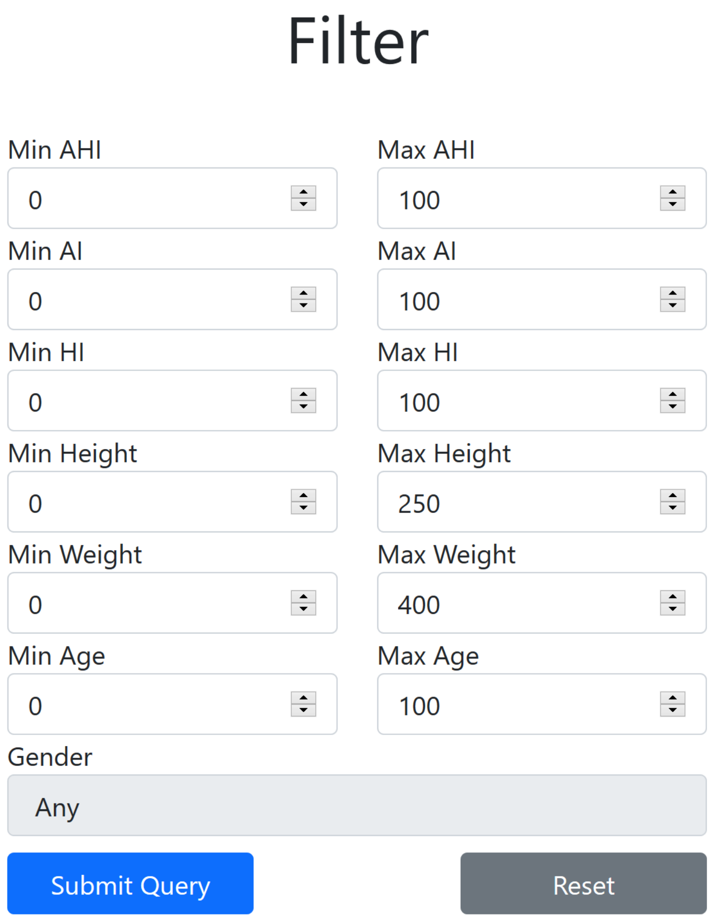
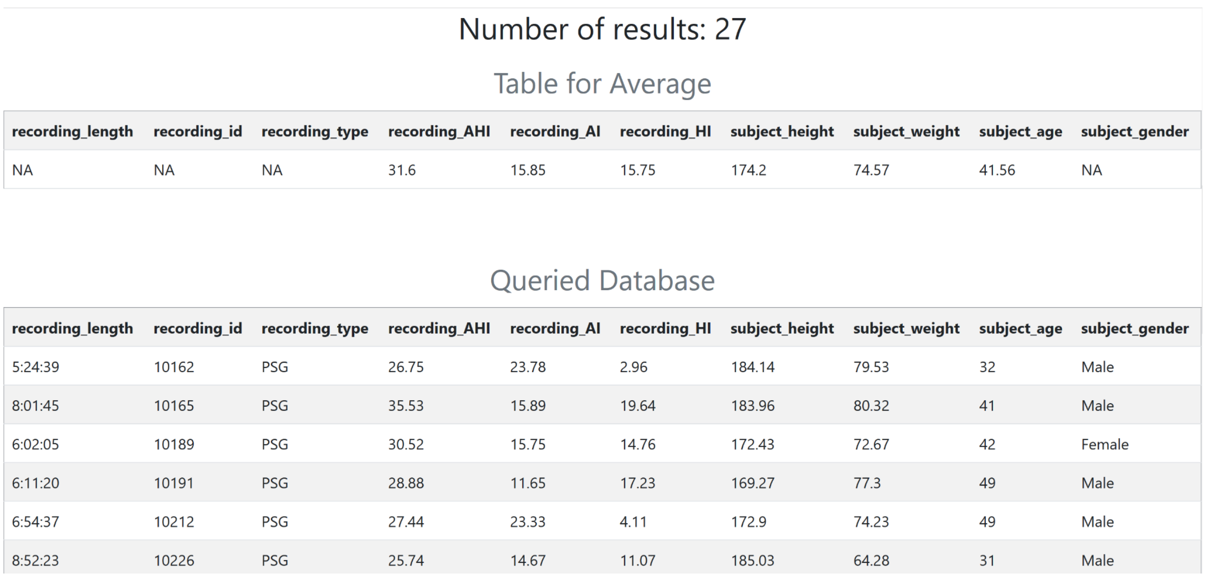


Figure S2: The interactable filter where the user can use to search the database for results.



Number of results: 27

Table for Average

recording_length	recording_id	recording_type	recording_AHI	recording_AI	recording_HI	subject_height	subject_weight	subject_age	subject_gender
NA	NA	NA	31.6	15.85	15.75	174.2	74.57	41.56	NA

Queried Database

recording_length	recording_id	recording_type	recording_AHI	recording_AI	recording_HI	subject_height	subject_weight	subject_age	subject_gender
5:24:39	10162	PSG	26.75	23.78	2.96	184.14	79.53	32	Male
8:01:45	10165	PSG	35.53	15.89	19.64	183.96	80.32	41	Male
6:02:05	10189	PSG	30.52	15.75	14.76	172.43	72.67	42	Female
6:11:20	10191	PSG	28.88	11.65	17.23	169.27	77.3	49	Male
6:54:37	10212	PSG	27.44	23.33	4.11	172.9	74.23	49	Male
8:52:23	10226	PSG	25.74	14.67	11.07	185.03	64.28	31	Male

Figure S3: The results after a filter has been used. The following shows, number of data entries, a table with the average of each column, and the filtered database.