CTU
CZECH TECHNICAL UNIVERSITY IN PRAGUE

F3

Faculty of Electrical Engineering
Department of Computer Science

Master's Thesis

Behavioral data processing

Bc. Lukáš Sláma
Open Informatics - Software Engineering

May 2024
Supervisor: doc. Ing. Daniel Novák Ph.D.

Acknowledgement

I would like to thank my supervisor Daniel Novák for his help and guidance. I would also like to thank Jakub Schneider for his help and counsel. Lastly I want to thank my brother Tomáš for his notes and suggestions and my whole family for their continued support.

Declaration

I declare that the presented work was developed independently and that I have listed all sources of information used within it in accordance with the methodical instructions for observing the ethical principles in the preparation of university theses.

In Prague, 24 May 2024

. . . .

Abstrakt / Abstract

Cílem této diplomové práce bylo prozkoumat obor behaviorálních dat a jeho aktuální využití v praxi.

Je popsána open-source platforma pro sběr behaviorálních dat, nazvaná LAMP, a její nedostatky. K odstranění těchto nedostatků byla vyvinuta full-stack webová aplikace, nazvaná Qualix. Tuto aplikaci lze využít k rozšíření funkčnosti platformy LAMP. K novým funkcím patří e-mailové oznámení, kontroly kvality dat a rychlé vizualizace.

V závěrečné části této práce popisuji studii, ve které jsem se za pomocí získaných behaviorálních dat účastníků pokusil o klasifikaci dnů na víkendy a pracovní dny pomocí metody podpůrných vektorů.

Výsledky studie ukazují, že u některých účastníků lze pomocí metody podpůrných vektorů odhalit vzorce chování, které mohou klasifikovat dny na víkendy a všední dny. U jiných účastníků ale byla přesnost modelů nízká a u některých nebyl shromážděn dostatek dat pro provedení klasifikace. To poukazuje na potřebu zlepšení postupů kontroly kvality dat, například využitím aplikace Qualix.

Klíčová slova: analýza, klasifikace bahviorálních dat; vývoj aplikace; rozpoznávání volných/pracovních dnů.

The aim of this master's thesis was to explore the field of behavioral data and its current use in practice.

An open-source platform for behavioral data collection, called LAMP, is described, along with its limitations. To address these shortcomings, a web-based full-stack application, called Qualix, was developed. This application can be utilized to provide additional features to the LAMP platform, including email notifications, data quality checks, and fast visualizations.

In the final section of this thesis, I present a study in which I collected behavioral data from participants. The objective of this study was to utilize the data to attempt classification of days into weekends/weekdays using support vector machines.

The results of the study indicate that for some participants, support vector machines can be employed to detect behavioral patterns that can classify days into weekends and weekdays. However, for other participants, the accuracy of the models was low, and for yet others, insufficient data was collected and classification was not possible. This highlights the need for improved data quality control procedures, such as the utilization of the Qualix application.

Keywords: behavioral data analysis, classification; application development, weekday/weekend pattern recognition.

Contents /

Tables / Figures

Chapter 1
Introduction

1.1 Behavior

Behavior can be defined as *"anything that an organism does involving action and response to stimulation"* [1]. For the purpose of this thesis, we can substitute *organism* for *human*. An important feature of behavior is that it is relational [2]. This means that behavior is not just a result of internal factors, such as genes, but also a result of external factors, such as the environment in which the organism is situated [2].

1.2 Behavioral data

1.2.1 Definition and introduction

Behavioral data refers to information collected through technology, such as computers, internet browsers, smartphones, smartwatches, smart rings, etc. about a person's behavior [3]. This data can be obtained passively from various sensors, such as from the accelerometer, GPS, gyroscope, microphone or screen, or actively by the user itself, such as through questionnaires [3].

We can use the data from these sensors to infer more complex information, such as a person's emotional state, mood or stress level. [4]

1.2.2 Inferring information from behavioral data

Wang et al. show that by looking at behavioral patterns of students, we can obtain their perceived stress, depression, flourishing and loneliness levels. Certain behavioral patterns are positively correlated with academic performance, such as conversation duration and frequency, while others are negatively correlated with academic performance, such as the amount of movement around campus. [5]

An exploratory study by Saeb et al. shows that behavioral markers that are strongly related to depressive symptom severity can be deduced from behavioral data obtained by mobile phones. [6]

1.2.3 How behavioral data can manipulate user behavior

Apart from obtaining personal information, such as a person's mood, from behavioral data, this data can be used to change a person's behavior and opinions. Social media such as Facebook or Twitter are suitable for this, since advertisements can be microtargeted to specific groups of people. Apart from advertisement, social media influencers can be used to share behavior change messaging. [7]

1.2.3.1 Positive effects

Positive behavior modification is usually done voluntarily with the consent from the user.

For example, Napolitano et al. show that using Facebook and text messaging to provide weight-loss information to young adults can lead to successful, lasting weight-loss for highly engaged participants [8] using the Healthy Body Healthy U (HBHU) trial design [9].

According to Evans et al., social media can be used to increase vaccination rates in low -and middle-income countries (LMICs), specifically in Nigeria, by showing pro-vaccination social norms messages to the users. [10]

In a randomized controlled trial (RCT) by Bonar et al. on teenagers and young adults (16-24 year olds), a social media intervention including incentives was found to be effective in reducing drug use (excluding alcohol), drug consequences and cannabis-impaired driving. [11]

In a different study by Bonar et al. on adults (18-25 year olds), an 8-week motivational interviewing and cognitive-behavioral intervention for cannabis use was carried out on social media. This RCT showed absolute reductions over time in several measures of cannabis consumption, such as total days used, total times used, total quantity used, for many ways of administration (such as smoking, vaping, consuming edibles, etc.) and others. [12]

1.2.3.2 Negative effects

Negative behavior change on the other hand is usually done involuntarily, without explicit consent from the user and sometimes even without the knowledge of the manipulated user.

Since current social media work on the model of surveillance capitalism, which is defined as *“an economic system in which the product for sale is personal data collected on the internet, ...”* [13] it is in the best interest of social media companies to gather as much data as possible, since this makes their targeted advertising better, which tends to increase their revenue [14]. To achieve this, social media companies employ numerous design elements to try and make their products as addictive as possible [15].

One of these elements is known as the *“endless scroll”* or *“endless stream”*. Endless scrolling produces a feeling known as *“flow”* which is characterized by strong concentration, pleasure and time distortion. These effects are very useful to social media companies, since a person can become so immersed in their product that they forget about time and space. Furthermore, since there is no natural stop to this activity, users don’t have a chance to evaluate their behavior and maybe think about turning off their phone/computer and doing something more suitable. [14]

Another element that is often used on social media sites is known as a *“temporal event”*. A temporal event is an event or an opportunity that only occurs inside a specific time period. An example of this is a feature of Instagram called *“stories”*. Stories are pictures and/or videos added by users which have a limited life-span, after which they are removed and are no longer accessible. This creates a fear of missing out which compels users to check for new stories frequently. [16]

The design element that is in my opinion the most dangerous is called *“intermittent reward schedule”*, which is a concept first discovered by B.F. Skinner [17]. Intermittent rewards work by having an element of uncertainty, in which a user of a social media platform (or a gambling addict playing the slot machine, or a rat pressing a lever to get food) doesn’t know what will come next. Uncertainty is determined by probability. When probability of an event is equal to 0.5, uncertainty is at a maximum. Likely due to evolutionary reasons, dopamine neurons activate when there is a possible reward with maximum uncertainty [18]. In the context of social media use, not all posts are interesting and therefore not all posts produce a reward in the pleasure centers of the brain. It is beneficial for social media companies to balance the amount of interesting and non-interesting posts to achieve maximum uncertainty for the user. This uncertainty is what is keeping users engaged in anticipation of the next interesting post. [19]

Furthermore, it is difficult to know how these design elements interact with each other and the resulting effect of these interactions on usage time. It may be possible that if one feature would increase usage time on average by 10 min and another one by 5 min, the resulting average increase of usage time when these two elements are used together would be far greater than the sum of individual elements, in this case 15 min. Unfortunately it is difficult to study these effects, since social media companies usually do not grant access to their data to independent scientists. [14]

From these design elements, it is clear to see that social media companies manipulate their users so that they will stay on their platforms for as long as possible. Other behavior modifications have been observed, for example Lindström et al. analyzed over one million social media posts from over 4000 individuals and came to the conclusion that social rewards (in the form of likes) influence users' behavior on social media and that users will strategically space out their posts in order to maximize social rewards [20].

User data can be very beneficial to social media companies, because they can then be used to deliver targeted advertising. Zarouali et al. show that user data can be used to infer whether a person is introverted or extraverted and based on this information deliver political micro-targeted messaging. In their study, Zarouali et al. found that these micro-targeted advertisements that match the users' personality traits are significantly more effective at persuading users than generic advertising or incorrectly targeted advertisements. [21]

A real-life example of the effectiveness of political micro-targeted advertisements is the Cambridge Analytica scandal. The company used, among other sources, private data of more than 87 million Facebook users without their knowledge. They used user data to deliver political micro-targeted advertisements to UK voters on behalf of a radical *Leave.EU* Brexit campaign and to US voters during the 2016 United States presidential election on behalf of Ted Cruz and later Donald Trump. It is unclear how effective this micro-targeted advertisement was, but Ted Cruz saw an unexpected rise in the primaries and statistical models predicted a sweeping loss for Donald Trump and yet the exact opposite had happened. As, at the time Cambridge Analytica CEO Alexander Nix explained, Cambridge Analytica was *"...able to form a model to predict the personality of every single adult in the United States of America."* [22]. This model was used not only to increase the number of votes for Ted Cruz and Donald Trump, but also to discourage potential Clinton voters from casting a vote in the election. [22–26]

While some users deleted their Facebook accounts after the scandal broke [27], the overall amount of users wasn't affected and in fact only increased [28]. Even though nearly two-thirds (65%) of social media users were familiar with the Cambridge Analytica scandal and nearly half (44%) of users viewed Facebook more negatively after the scandal, according to a survey from 2019 done by The Manifest [29]. The reason for user apathy could be explained by a study by Hinds et al. in which they interviewed 30 participants based at a UK university. While 26 out of 30 participants knew about the scandal, many believed that they are immune to psycho-graphically targeted advertisements, even though they often lack understanding of how the algorithms behind the advertisements work. One explanation could be related to cognitive dissonance. Users like using Facebook and they want to keep using it in the future. When they become aware of privacy risks or data leaks, they can either change their behavior (e.g. stop using the platform) or change their opinions about the potential risks (e.g. by adapting a belief that they are immune to micro-targeted advertisements). As seen by the platforms' usage statistics, it would seem that the latter option is more popular. This is just one possible explanation however and it is likely that a more complicated processes guide the behavior of users that choose to stay on Facebook even after becoming aware of Facebooks misuse of their personal data. [30]

Thankfully, all of these negative (and positive) effects of behavioral data can be prevented by users themselves. Ad-blockers can be used to remove targeted advertisements and strengthening privacy settings can be used to revoke privileges to certain personal data.

As for the addictive elements, users can make a choice, based on how much they are affected by these elements, to abstain from social media or to limit their use. For users that cannot control their use anymore, some promising treatment approaches are emerging. Current research shows that the most efficacious way of treating internet addiction (IA) is by using repetitive trans-cranial magnetic stimulation (rTMS) and cognitive behavioral therapy (CBT), or a combination of both. [31–32]

1.2.4 Importance of behavioral data quality

As Currey and Torous explain, data coverage for behavioral data is rarely 100% [33]. Such a high coverage is often not necessary, but it is nevertheless important to keep it as high as possible in order to achieve accuracy of derived features. If we wanted to classify significant locations of users based on their GPS data from their smartphones for example, low data coverage could lead to incorrectly classifying a users' home address and/or time spent at certain locations. This could negatively impact clinical interpretations. [33]

1.2.5 Summary

As with any other technology, behavioral data are neutral and can be used to bring about both positive and negative effects on people. The purpose of this work is not to theorize whether or not behavioral data should be used or not, whether they should be more regulated or even outright banned, but to show that for behavioral data to bring about a desired effect, the quality of such data is of utmost importance.

As I've described in the previous sections, behavioral data from social networks can be used to infer information about its participants and then to modify their behavior. In this thesis I will focus mainly on behavior data gathered from smartphone sensors, since they don't require any input from the user, unlike behavioral data from social media. Several platforms for behavioral data acquisition exist, one of them being the LAMP platform.

1.2.6 LAMP platform for behavioral data acquisition

The LAMP (Learn, Assess, Manage and Prevent) platform was developed by a team at the *Division of Digital Psychiatry* at the *Beth Israel Deaconess Medical Center*, which is a Harvard Medical School affiliate in Boston, MA. The platform was first released in 2019 and has been receiveing updates ever since [34–36]. The code is available online[1] under a BSD-3 license. The platform has 4 main parts:

[1] `https://github.com/BIDMCDigitalPsychiatry/LAMP-platform`

- Smartphone app
- Web based Dashboard
- Database
- CORTEX package for data analysis

A more complex description of the platform and its features is available in my bachelor's thesis [37].

In this thesis, I'm focusing mainly on the Dashboard and CORTEX parts. We've tested the platform thoroughly over several years and have come to the conclusion that there are various shortcomings. The Dashboard uses large UI elements to show information about the quality of patients' data, which makes it difficult to assess them at a glance. Moreover, there is no history of data quality present. As for the CORTEX package, it comes in the form of a Python library and it makes generating visualizations cumbersome, since the user first has to describe the desired visualizations in code, then execute the code and then use the Dashboard to find and download the generated visualization.

That is why I have developed a full-stack web based application called *Qualix* that aims to improve the shortcomings of the LAMP platform and add non-existing features.

Chapter 2
Qualix application

2.1 Introduction

Qualix is a web-based application that uses Java, Spring Boot and PostgreSQL as its main technologies and Vaadin for its graphical user interface (GUI). For data processing and visualization, Python with pandas and matplotlib libraries were used. The code is freely available[1] under a GPLv3[2] license. Qualix is designed to run on Docker containers in order to be easy to deploy (for the build and deployment guide, see D).

2.2 Feature overview

There are 4 main features of Qualix:

1. Storing the participants of a LAMP study, their contact information and enrollment status. Since the LAMP platform itself doesn't have a feature of storing the study participant information or their enrollment status, I've implemented this feature in the List view. This view starts out empty — image 2.3, and shows rows for Study Id, First Name, Last Name, Email, Phone Number and Enrollment Status. After it connects to the LAMP platform and downloads the IDs of study participants, it will generate placeholder information — image 2.4, since this information is not stored on the LAMP server. The study staff can then choose whether they want to use this feature or not and based on this decision, change the placeholder information to real data — images 2.5 2.6.

Since this view can contain personal information, only one of the two types (admin, user) of Qualix users — admins, can access this view. For the privacy of study participants, Qualix user privileges can be used to allow only some of the study staff access to this view.

2. Automatic data quality check. For the automatic data quality check a `@Scheduled` method[3] is used, which is set up to run every day at 7 AM. This time was chosen because the data check can take some time, based on the amount of participants in a given study. If the study staff wanted to check the data quality at a later time, in the evening for example, a different time can be set. It is important to choose a time for the duration of the study so that the data quality checks are consistent.

The data quality check result can be seen in the Dashboard view, which will be empty after first starting the application — image 2.7. After connecting to the LAMP server, placeholder information in the form of black dots will be generated — image 2.8. After that, Qualix will check when it last received data from each sensor and color the black dots according to this rule — image 2.9:

[1] `https://github.com/lukislama/Qualix`

[2] `https://www.gnu.org/licenses/gpl-3.0.en.html`

[3] It is the `getDataQualityForPreviousDay()` method of the *MainService.java* class described at 2.3.3

- Green — Latest data arrived in the last 2 hours
- Yellow — Latest data arrived in the last 12 hours
- Red — Latest data arrived more than 12 hours ago
- Black — no data at all

Qualix also keeps a history of data quality check for every day. Clicking on a participant's row will show the history of that participant — image 2.10.

3. Email notifications regarding the data quality. After the automatic data quality check, if the notification email is set, Qualix will send a data quality overview via email — image 2.11.

4. Fast data processing and visualization thanks to a data and visualization cache. The last features are data caches and data processing and visualization. The data cache first needs to be created, which can be done in the Settings view via the *Create cache* button — image 2.12. Since this data cache always keeps the last 7 days of LAMP data locally, it needs to download and process all of this data when first being generated. This can take several hours depending on the amount of data and the number of participants in a given study. After the data cache is generated, a visualization cache is generated next. The application processes the data of all participants and generates a visualization for each feature for 1, 3, 5, and 7 days of data. This is desirable since when a member of a study staff needs to generate some visualization, it is instant, since all of the data have been processed and visualized in the background.

Data cache consolidation works by deleting the 8th day of data and downloading the 1st from the LAMP server. It also deletes old visualizations and generates new ones. It is handled by a `@Scheduled` method[4] which is set up to run every night at 3 AM. This time was chosen because at this time it is very likely that all of the data from the previous day have already been uploaded to the LAMP server and there is enough time until the morning when all visualizations should be ready to use.

Visualizations can be generated in the *Visualizations* tab. When selecting a visualization length, the options 1, 3, 5, and 7 days are using cached visualizations and will be generated instantly — images 2.13 2.14 2.15 2.16. If the custom option is selected, the desired range of data will be downloaded from the LAMP server, processed and visualized — images 2.17 2.18. This can take some time, especially when generating features that work with accelerometer data, since accelerometer usually has a very high pull rate and a large amount of data needs to be processed.

2.3 Code overview

2.3.1 Selecting a platform

When I first started developing the application, it was my intention to use a full-stack platform that was wholly based on Java and where the programmer would be able to program the entirety of the UI directly within the Java code. For this reason I tried Vaadin. I used the *Vaadin flow-crm-tutorial*, which is available on GitHub[5] and is licensed under a *Unlicense*[6] license and is therefore free, unencumbered and released into the public domain, as a base code for Qualix and I have used some of its code in the final application. The reason for this is that I haven't worked with Vaadin before

[4] It is the `consolidateDataCache()` method of the *MainService.java* class described at 2.3.3

[5] `https://github.com/vaadin/flow-crm-tutorial`

[6] `https://unlicense.org/`

(or any other frontend Java platform for web applications for that matter) and I needed to familiarize myself with it. After coming to the conclusion that the Vaadin platform was suitable for my application, I realized that some parts of the tutorial, such as the Dashboard view or the Login view, would fit perfectly for use in my application, so I adapted them slightly to fit my needs.

2.3.2 Backend overview

The backend of Qualix consists of three main parts:

1. Java and Spring Boot server side
2. PostgreSQL database
3. Python scripts

The Java and Spring Boot part takes care of how the application operates and the Python part handles connection to the LAMP server and visualizations. The PostgreSQL database is used for storing the information of participants and their data quality history.

2.3.3 Java and Spring Boot server side

The application structure utilizes a modified version of the model, view, controller (MVC) architecture. This architecture separates the classes of a program into three parts [38]:

- Model, which encapsulates information
- View, which displays information to the user
- Controller, which implements actions

I was inspired by this architectural pattern and separated the application into Security, View and Data, while Data is further separated into Entity, Repository and Service. A class diagram of the Data package — image 2.1, shows relationships between its classes.

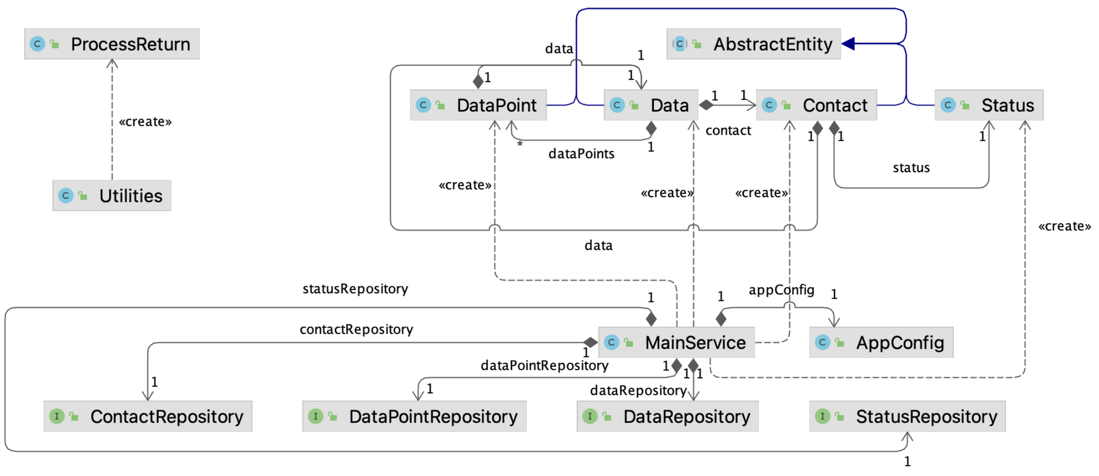


Figure 2.1. Data package diagram with relationships between classes.

The *MainService.java* can be thought of as the Controller in the MVC pattern. When Qualix is deployed for the first time, the MainService class checks whether there is any login information set in the *application.properties* file and if so, it connects to the LAMP server and downloads the IDs of participants in a set study. Similarly, it checks whether or not an email notification information is set and if so, sends a notification email to the set recipient that reads:

```
Receiving email set.
You email has been set to receive notifications from the Qualix
application.
If you think this is a mistake, please contact the study administrators
at SET STUDY ADMIN EMAIL.
```

where `SET STUDY ADMIN EMAIL` is the Google Mail sender address set in *application.properties*.

There are two methods in the MainService class with the `@Scheduled` annotation. These are `getDataQualityForPreviousDay()` and `consolidateDataCache()`. These methods will therefore run every day without needing any input from the user. The `getDataQualityForPreviousDay()` will connect to the LAMP server every morning and check the data quality from the previous day. It will then save this information to the PostgreSQL database and if an email is set, send a notification email with the data quality information. The `consolidateDataCache()` method runs every night and deletes the last day of the data and visualization cache and replaces it with a new day. Explanation of how these methods fit in with the features of Qualix can be found in 2.2.

The *AppConfig.java* class is used to load the relevant information from the *application.properties* file using the `@ConfigurationProperties` annotation with an annotation parameter `prefix = 'qualix'`. All of the relevant information is then stored in this class including the status of the cache and the supported visualization types. In the MainService class, only one private final variable of the AppConfig class is kept.

The *Utilities.java* class is a “static” class which cannot be instantiated. There are two static methods — `createAndRunProcess(String ... args)` and `sendEmail(args)` and one static class *TableBuilder*. The first method also uses the *ProcessReturn.java* class and is used to create and run processes using the Java ProcessBuilder class. This method is used to run the Python scripts from inside the Java application. More on these scripts in 2.3.4. The second method is used to send the daily notification emails regarding the data quality. Currently, this method is setup to only work with a Google account. This decision was made because Google is the only[7] large email provider left, that still allows this type of email sending for free. Using the Google app passwords[8] it is possible to connect to a Google account without having to use the Google account password itself and without using any proprietary libraries from Google. Using the app password, it was possible to use the native jakarta libraries for sending notification emails. The TableBuilder class is available online[9] under a Unlicense license. It is used to create a text-based table that encapsulates the data quality information that is sent via email.

As for the classes with an `@Entity` annotation, the *Contact.java* class is used to store information about a given participant. It has a ManyToOne relationship with the *Status.java* class, which stores a single String representing the study status of a participant, and a OneToOne relationship with the *Data.java* class. The Data class stores information about the data quality for each sensor (GPS, Accelerometer, Display

[7] To the authors’ knowledge.
[8] `https://support.google.com/accounts/answer/185833?hl=en&sjid=3303992743476717718-EU`
[9] `https://gist.github.com/JohnnyJayJay/3f7e85b3f5e52286d9f2ad6fbf17e3a6`

and Device motion). Apart from the OneToOne relationship with the Data class, it has a OneToMany relationship with the *DataPoint.java* class, which stores the history of data quality for each sensor — one DataPoint object is used for one day of data. Each Entity class also has its correspodning Repository interface, which handles the communication with the PostgreSQL database.

Next for the Security package, the *SecurityService.java* class interacts with the Spring Security framework and is used to get the current user's login details and to programatically log out of the application. The *SecurityConfig.java* class enables Spring Security and creates two default users — one with a username and password set to "user" with a role of "USER" and the other with a username and password set to "admin" with a role of "ADMIN". The class also allows access to all GET requests of */images/*.png* without authorization[10].

As for the Views package, a class diagram 2.2 shows the relationships between the classes.

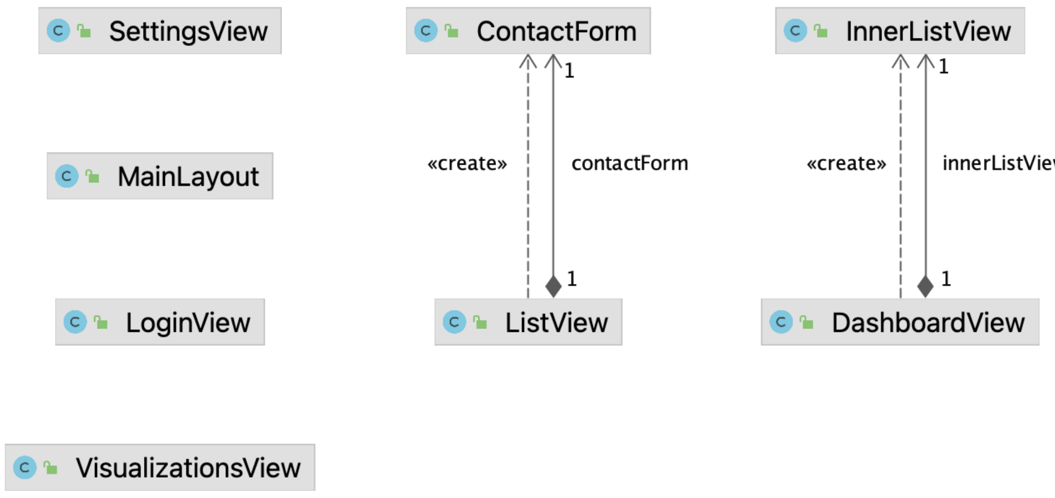


Figure 2.2. Views package diagram with relationships between classes.

All of the main views — Dashboard, List, Settings and Visualizations, have as a parent the *MainLayout.java* class. This class specifies the header with the application name, welcome sign and a Log out button, and the navigation drawer which is used to navigate to other views. The *LoginView.java* class is the first view a user sees when using the application. After logging in, the user will see the Dashboard view. The Dashboard view has a grid inside, which shows the latest data quality information for each patient. Inside the Dashboard view is the *InnerListView.java* class, which is used to display the history of data quality for a given participant. This view is only visible when the user clicks on a given participant in the Dashboard view. When that happens, the grid in the InnerListView will show up and since the grid in the Dashboard view and the whole InnerListView are in a HorizontalLayout, the two grids will appear next to each other with a 50/50 split in width.

The *ListView.java* class is only visible to users with the role "ADMIN" using the `@RolesAllowed` annotation and it also utilizes a Grid object to present information

[10] This is done because the *offlinePath* and *offlineResources* annotation parameters of the `@PWA` annotation in the *Application.java* class are used to fetch a .html page with a .png image located in *./images/*.

about the study participants. The ListView class also instantiates a ContactForm class, which is used to modify the information of a given participant. Similarly to the Dashboard view, the ContactForm is only visible when a specific user is selected in the Grid and is in a HorizontalLayout with the main Grid object. The split in width is 2/1 favoring the Grid.

The *SettingsView.java* class is also only accessible to users with the “ADMIN” role and is used to modify the LAMP server information, Google account information and to manually create the data and visualization caches. This class uses the Utilities class to create and run Python scripts that communicate with the LAMP server. More on these Python scripts at 2.3.4.

The *VisualizationsView.java* is accessible to all users and is used to create and display visualizations of collected data. This is also done with the help of Python scripts. If a user chooses to create a visualization using the data cache, the image will just be pulled up from the visualization cache and inserted as a simple .png image onto the page. The user can then download the image by right clicking on it and saving the image. In the case that the visualization cannot be pulled up from the visualization cache, a python script is called and this script will go through the data cache and generate the visualization. This will take up to a few minutes, but it is still much faster than downloading the data from LAMP and not using the data cache. In any case, the visualization cache is automatically consolidating itself every day, so a situation like this is unlikely to happen.

2.3.4 Python scripts

Qualix uses Python for connecting to the LAMP server and to process participants' data and create visualizations. It is not possible to use Java to directly communicate with the LAMP server. The documentation[11] provides three ways: JavaScript, Python and R. While JavaScript is technically supported according to the documentation, there is however no mention in it on how to actually use it to connect to LAMP and communicate with it. That left R and Python and I made the decision in favor of the latter, since it is a language with which I am more familiar. As for the visualizations, Java is not very well equipped to create visualizations easily in my opinion, so there would be a need to create and run a different process anyway, and since Python is already used to communicate with LAMP, it is logical to use it for data processing and visualization as well. Apart from this, a powerful and easy to use library for data visualization called matplotlib exists and was used to generate the visualizations for Qualix [39]. The following scripts are used to communicate with the LAMP server.

- *test_connection.py* tries to connect to the LAMP server and if it succeeds, returns an exit code of 0. If a connection is unsuccessful, the LAMP library will throw an exception and exit with a non-zero exit code.
- *get_study_participants.py* tries to connect to the LAMP server and if successful, it returns a list of the IDs of participants in the given study.
- *get_participant_last_data_time.py* tries to connect to the LAMP server and if successful, it returns a timestamp of the time when a latest row of data was received for each sensor of a given participant.
- *download_and_visualize_data.py* tries to connect to the LAMP server and if successful, it will download the desired data for a desired participant and a desired sensor and after downloading, it will process and visualize the data. This is all done in memory and no files are created.

[11] `https://docs.lamp.digital/`

- *create_data_cache.py* tries to connect to the LAMP server and if successful, it will download a week of data for all desired sensors for each participant and save them to .csv files with this naming scheme: *PARTICIPANT ID + SENSOR TYPE + START TIME + END TIME* where the start and end times are written in this scheme: *Y-m-dTH:M*. The resulting file can then look like this: *U3748598636Accelerometer2023-12-03T00:002023-12-04T00:00.csv*. This indicates that the file contains data for a participant with ID U3748598636, data type is Accelerometer and it contains data for the whole day of 3rd December 2023. One .csv file will be created for each sensor of each participant for each day of the week. This naming scheme is useful, because it is then easy to find the desired data for a given day/s quickly.
- *consolidate_data_cache.py* tries to connect to the LAMP server and if successful, it will download the latest day of data for all desired sensors for each participant and save them to the data cache using the standardized .csv file naming scheme. It then deletes the last day of data from the cache. This ensures that the data cache always has the last 7 days of data available.

The following scripts do not connect to the LAMP server and are used for visualizations:

- *download_and_visualize_data_from_cache.py* searches the data cache based on the input participant id, sensor type and visualization length. After searching for the correct files, it loads the data from it to memory, processes it and generates and saves a visualization. The visualizations also follow a naming scheme similar to the data cache: *PARTICIPANT ID + _ + VISUALIZATION TYPE + _ + START TIME + _ + END TIME*. The start and end times follow the same scheme as the data cache. The reason for the underscores in the visualization cache is so that a user looking through the files can immediately recognize what file is a data cache file and what file is a visualization file without having to look at the file extension.
- *delete_old_visualizations.py* is a simple script that deletes all visualizations from the visualization cache. This is necessary since after consolidating the data cache, these visualizations are no longer relevant since they don't contain the latest day of data.

2.3.5 System architecture and PostgreSQL database

Qualix is designed to run on Docker containers. Two containers are generated by the *docker-compose.yml* file for this project. The first is a linux image with the latest Python version installed. The necessary libraries are then installed using pip. To this image the JDK from eclipse-temurin:21 is copied so that the Qualix Java application can run on it. The other container is a postgres:16.1 image, which is used as the database for Qualix. Docker will automatically connect both containers to the same internal network and then Qualix will connect to the database according to the information within the docker compose file.

2.4 Comparison with the LAMP platform

As stated in the introduction — 1.2.6, Qualix was developed to complement the LAMP platform and provide some missing features. The LAMP Dashboard — image 2.19, is the main control panel for the LAMP platform. Using it, users can see information about when was the last data received, but without any history. The UI elements are large and therefore not that many participants fit in a single page, which makes checking data quality at a glance difficult. There is also no possibility of receiving automatic notification emails regarding the data quality.

As for the visualization part, LAMP comes with CORTEX[12], which is a data analysis and visualization toolkit. The CORTEX itself is a very nice toolkit in my opinion, offering a wide range of pre-programmed data analysis and visualization functions. This functionality is kneecapped however by the way users are forced to use it. The user first has to install CORTEX as a Python library. The documentation mentions that any Python 3.4+ will suffice. In my testing however, I was only able to get CORTEX running on Python 3.8. After the installation is finished, the user has to write a Python script in order for the data processing and/or visualization to take place. After that, the data will be processed and visualized. The user cannot access the visualizations just yet however, since they are stored in the Data Portal section of the LAMP Dashboard. In my testing, the Data Portal is very buggy, hangs often and utilizes strange UI elements — images 2.20 2.21 2.22 2.23 2.24. This process of visualization generation is very time consuming and complicated. Compared with Qualix, where visualizations are generated using simple UI elements and often instantly, it is a much worse user experience.

2.5 Screenshots

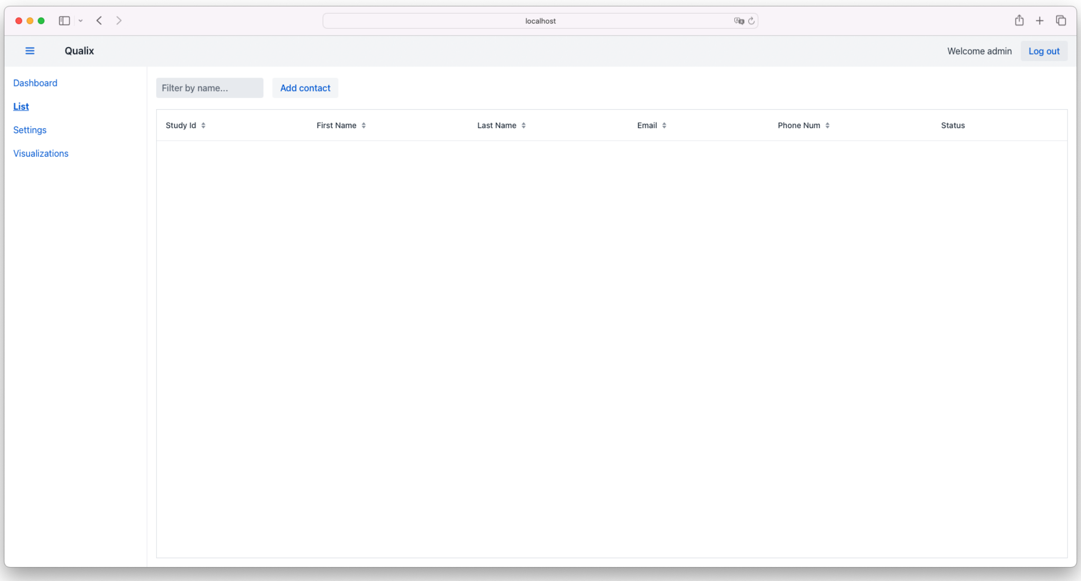
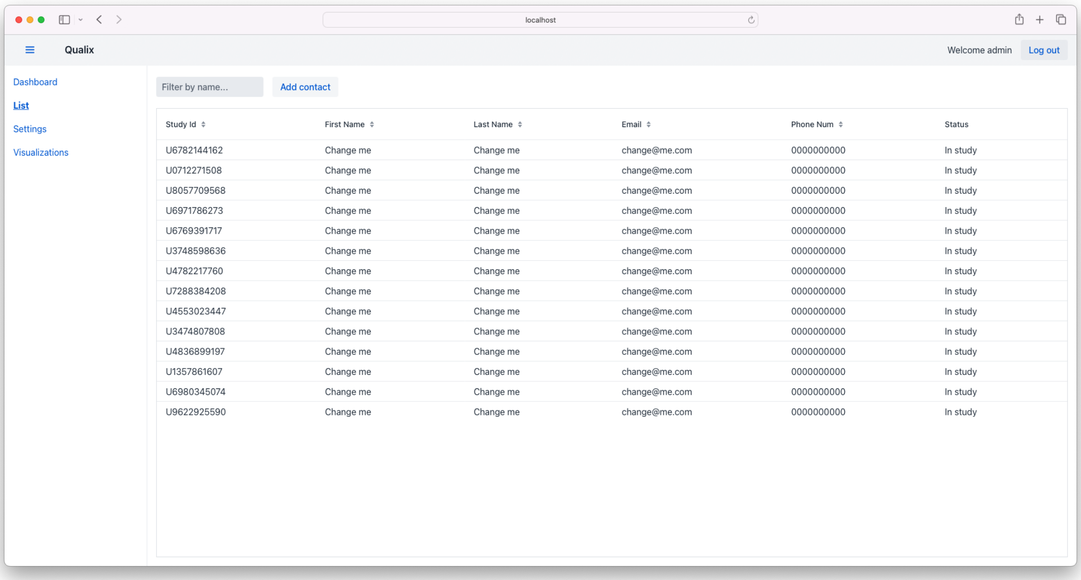


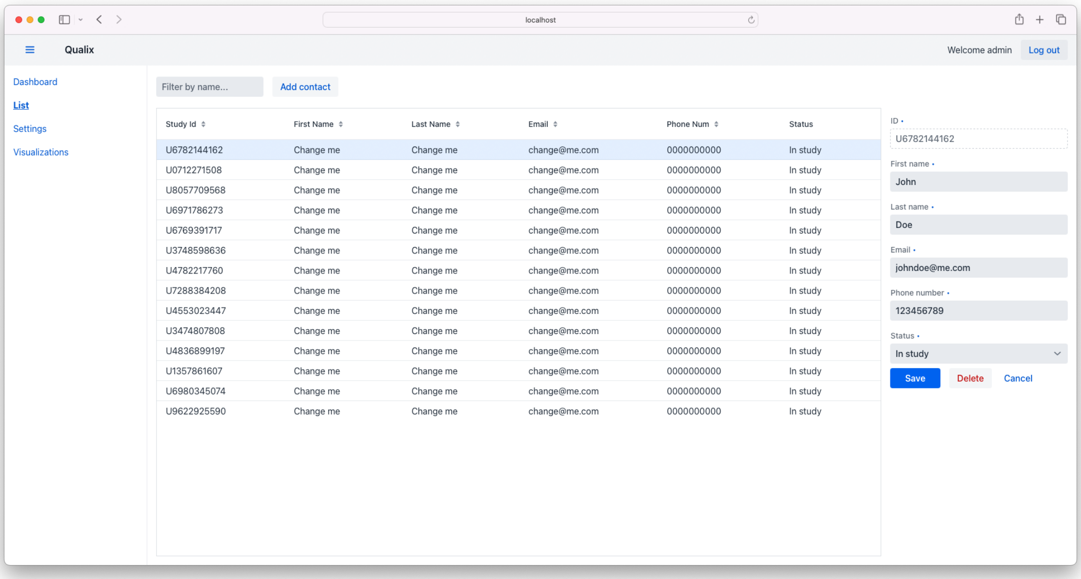
Figure 2.3. Qualix Empty List view.

[12] https://docs.lamp.digital/data_science/cortex/what_is_cortex



Study Id	First Name	Last Name	Email	Phone Num	Status
U6782144162	Change me	Change me	change@me.com	0000000000	In study
U0712271508	Change me	Change me	change@me.com	0000000000	In study
U8057709568	Change me	Change me	change@me.com	0000000000	In study
U6971786273	Change me	Change me	change@me.com	0000000000	In study
U6769391717	Change me	Change me	change@me.com	0000000000	In study
U3748598636	Change me	Change me	change@me.com	0000000000	In study
U4782217760	Change me	Change me	change@me.com	0000000000	In study
U7288384208	Change me	Change me	change@me.com	0000000000	In study
U4553023447	Change me	Change me	change@me.com	0000000000	In study
U3474807808	Change me	Change me	change@me.com	0000000000	In study
U4836899197	Change me	Change me	change@me.com	0000000000	In study
U1357861607	Change me	Change me	change@me.com	0000000000	In study
U6980345074	Change me	Change me	change@me.com	0000000000	In study
U9622925590	Change me	Change me	change@me.com	0000000000	In study

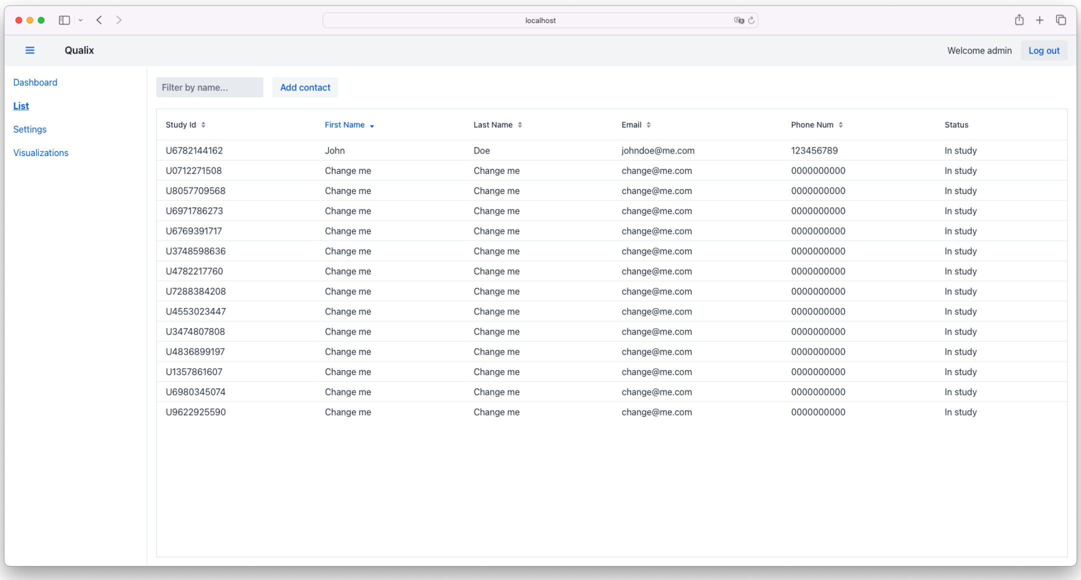
Figure 2.4. Qualix List view - Before info change.



Study Id	First Name	Last Name	Email	Phone Num	Status
U6782144162	Change me	Change me	change@me.com	0000000000	In study
U0712271508	Change me	Change me	change@me.com	0000000000	In study
U8057709568	Change me	Change me	change@me.com	0000000000	In study
U6971786273	Change me	Change me	change@me.com	0000000000	In study
U6769391717	Change me	Change me	change@me.com	0000000000	In study
U3748598636	Change me	Change me	change@me.com	0000000000	In study
U4782217760	Change me	Change me	change@me.com	0000000000	In study
U7288384208	Change me	Change me	change@me.com	0000000000	In study
U4553023447	Change me	Change me	change@me.com	0000000000	In study
U3474807808	Change me	Change me	change@me.com	0000000000	In study
U4836899197	Change me	Change me	change@me.com	0000000000	In study
U1357861607	Change me	Change me	change@me.com	0000000000	In study
U6980345074	Change me	Change me	change@me.com	0000000000	In study
U9622925590	Change me	Change me	change@me.com	0000000000	In study

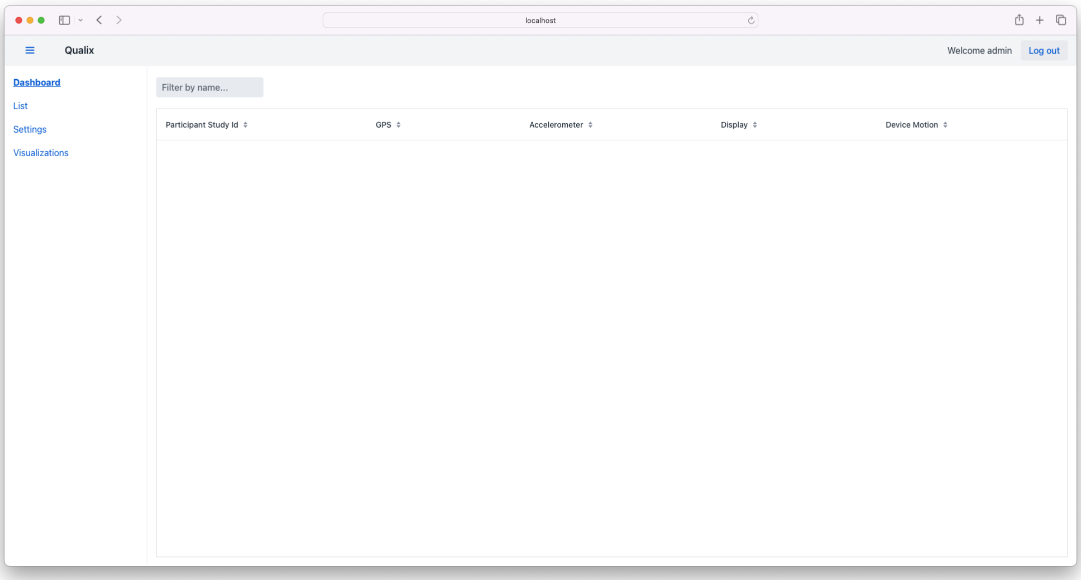
ID •
U6782144162
First name •
John
Last name •
Doe
Email •
johndoe@me.com
Phone number •
123456789
Status •
In study

Figure 2.5. Qualix List view - Changing info.



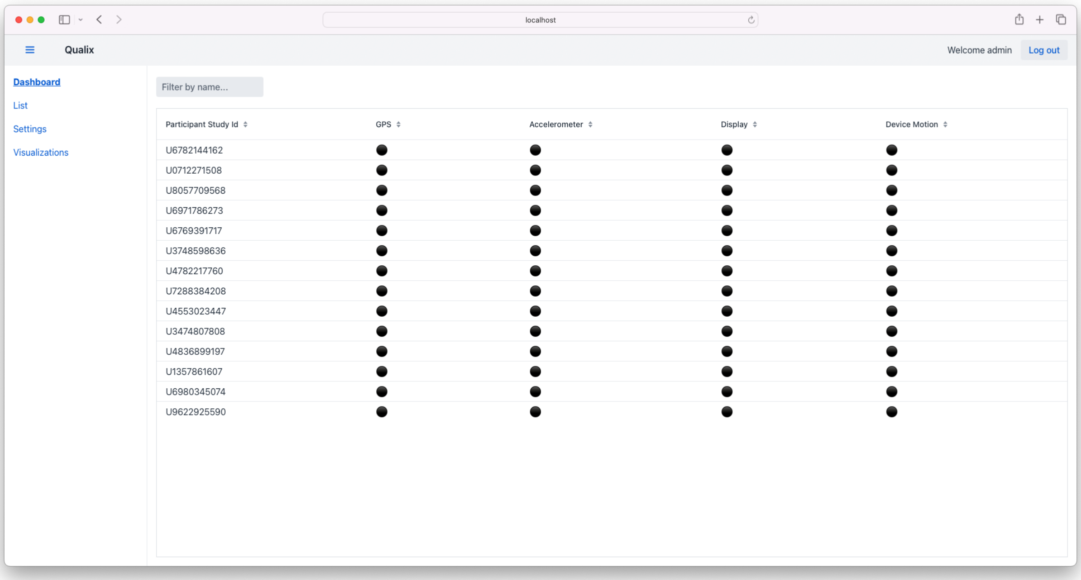
Study Id	First Name	Last Name	Email	Phone Num	Status
U6782144162	John	Doe	johndoe@me.com	123456789	In study
U0712271508	Change me	Change me	change@me.com	0000000000	In study
U8057709568	Change me	Change me	change@me.com	0000000000	In study
U6971786273	Change me	Change me	change@me.com	0000000000	In study
U6769391717	Change me	Change me	change@me.com	0000000000	In study
U3748598636	Change me	Change me	change@me.com	0000000000	In study
U4782217760	Change me	Change me	change@me.com	0000000000	In study
U7288384208	Change me	Change me	change@me.com	0000000000	In study
U4553023447	Change me	Change me	change@me.com	0000000000	In study
U3474807808	Change me	Change me	change@me.com	0000000000	In study
U4836899197	Change me	Change me	change@me.com	0000000000	In study
U1357861607	Change me	Change me	change@me.com	0000000000	In study
U6980345074	Change me	Change me	change@me.com	0000000000	In study
U9622925590	Change me	Change me	change@me.com	0000000000	In study

Figure 2.6. Qualix List view - After info change.



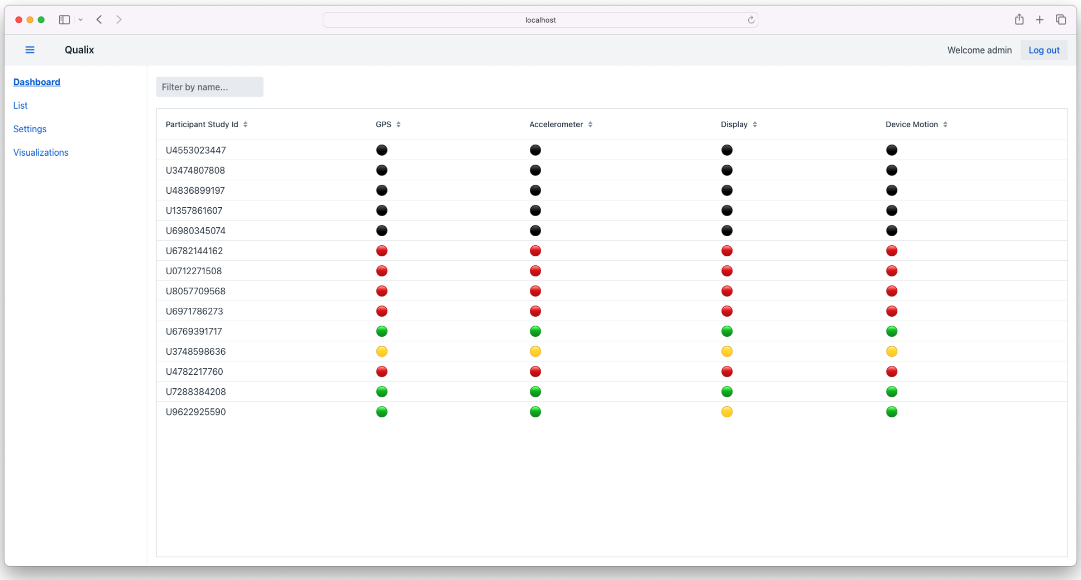
Participant Study Id	GPS	Accelerometer	Display	Device Motion

Figure 2.7. Qualix Empty Dashboard view.



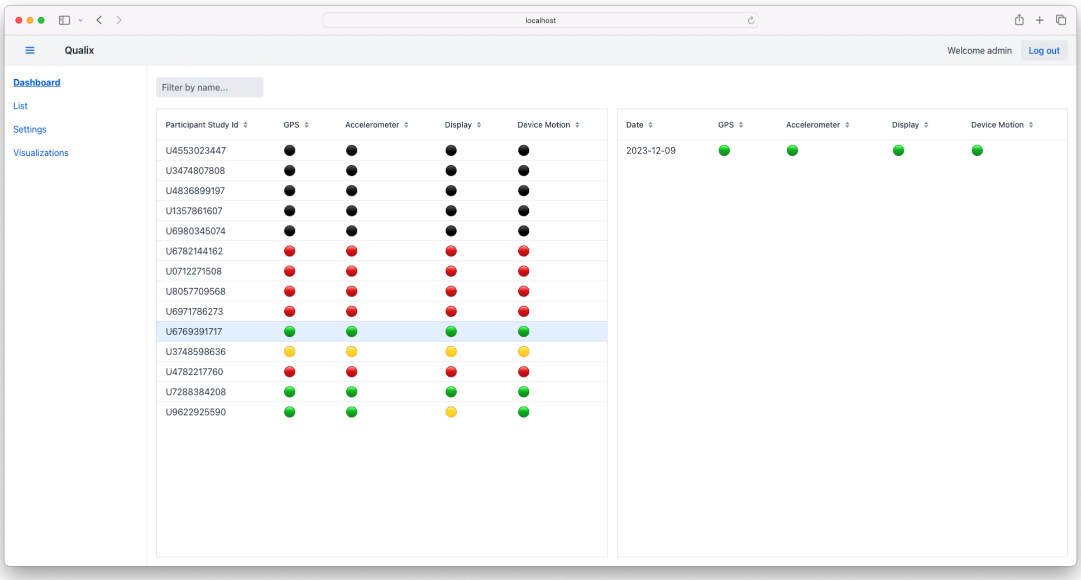
Participant Study Id	GPS	Accelerometer	Display	Device Motion
U6782144162	●	●	●	●
U0712271508	●	●	●	●
U8057709568	●	●	●	●
U6971786273	●	●	●	●
U6769391717	●	●	●	●
U3748598636	●	●	●	●
U4782217760	●	●	●	●
U7288384208	●	●	●	●
U4553023447	●	●	●	●
U3474807808	●	●	●	●
U4836899197	●	●	●	●
U1357861607	●	●	●	●
U6980345074	●	●	●	●
U9622925590	●	●	●	●

Figure 2.8. Qualix Dashboard view - Before data check.



Participant Study Id	GPS	Accelerometer	Display	Device Motion
U4553023447	●	●	●	●
U3474807808	●	●	●	●
U4836899197	●	●	●	●
U1357861607	●	●	●	●
U6980345074	●	●	●	●
U6782144162	🔴	🔴	🔴	🔴
U0712271508	🔴	🔴	🔴	🔴
U8057709568	🔴	🔴	🔴	🔴
U6971786273	🔴	🔴	🔴	🔴
U6769391717	🟢	🟢	🟢	🟢
U3748598636	🟡	🟡	🟡	🟡
U4782217760	🔴	🔴	🔴	🔴
U7288384208	🟢	🟢	🟢	🟢
U9622925590	🟢	🟢	🟡	🟢

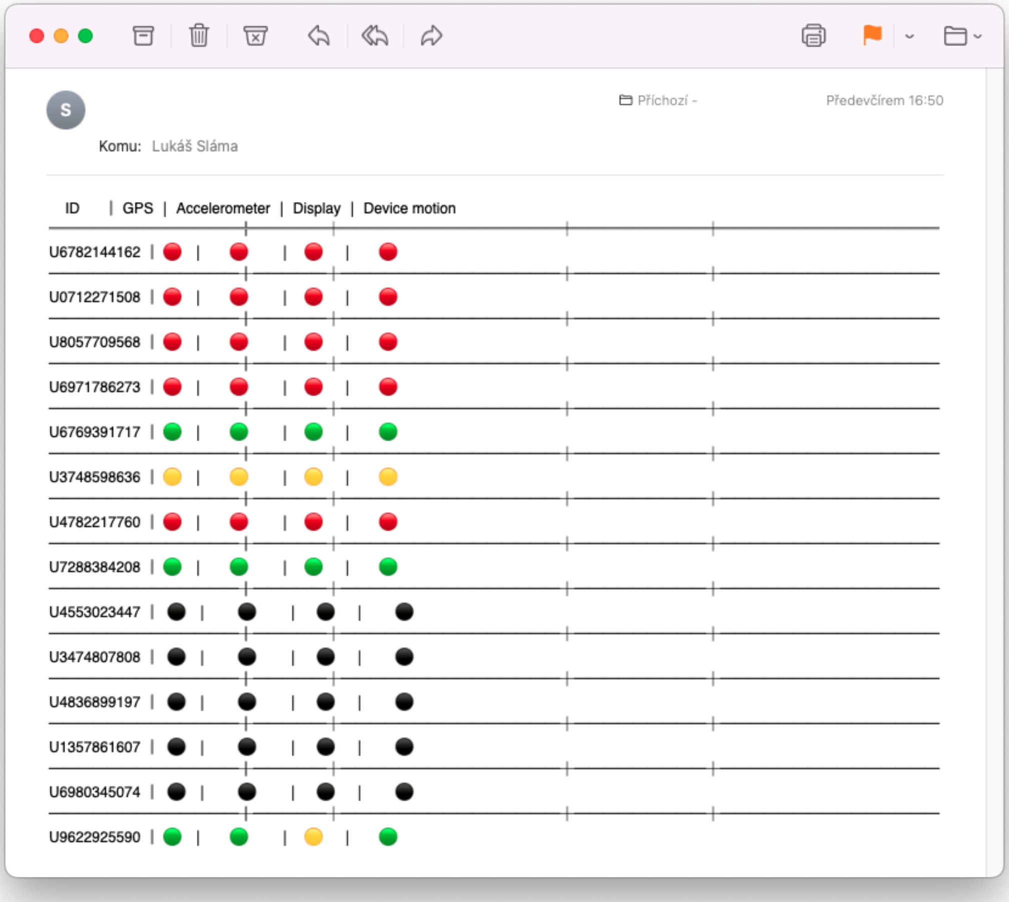
Figure 2.9. Qualix Dashboard view - After data check.



Participant Study Id	GPS	Accelerometer	Display	Device Motion
U4553023447	⚫	⚫	⚫	⚫
U3474807808	⚫	⚫	⚫	⚫
U4836899197	⚫	⚫	⚫	⚫
U1357861607	⚫	⚫	⚫	⚫
U6980345074	⚫	⚫	⚫	⚫
U6782144162	🔴	🔴	🔴	🔴
U0712271508	🔴	🔴	🔴	🔴
U8057709568	🔴	🔴	🔴	🔴
U6971786273	🔴	🔴	🔴	🔴
U6769391717	🟢	🟢	🟢	🟢
U3748598636	🟡	🟡	🟡	🟡
U4782217760	🔴	🔴	🔴	🔴
U7288384208	🟢	🟢	🟢	🟢
U9622925590	🟢	🟢	🟡	🟢

Date	GPS	Accelerometer	Display	Device Motion
2023-12-09	🟢	🟢	🟢	🟢

Figure 2.10. Qualix Dashboard view - More info.



Předevčírem 16:50

Komu: Lukáš Sláma

ID	GPS	Accelerometer	Display	Device motion
U6782144162	🔴	🔴	🔴	🔴
U0712271508	🔴	🔴	🔴	🔴
U8057709568	🔴	🔴	🔴	🔴
U6971786273	🔴	🔴	🔴	🔴
U6769391717	🟢	🟢	🟢	🟢
U3748598636	🟡	🟡	🟡	🟡
U4782217760	🔴	🔴	🔴	🔴
U7288384208	🟢	🟢	🟢	🟢
U4553023447	⚫	⚫	⚫	⚫
U3474807808	⚫	⚫	⚫	⚫
U4836899197	⚫	⚫	⚫	⚫
U1357861607	⚫	⚫	⚫	⚫
U6980345074	⚫	⚫	⚫	⚫
U9622925590	🟢	🟢	🟡	🟢

Figure 2.11. Qualix Email report.

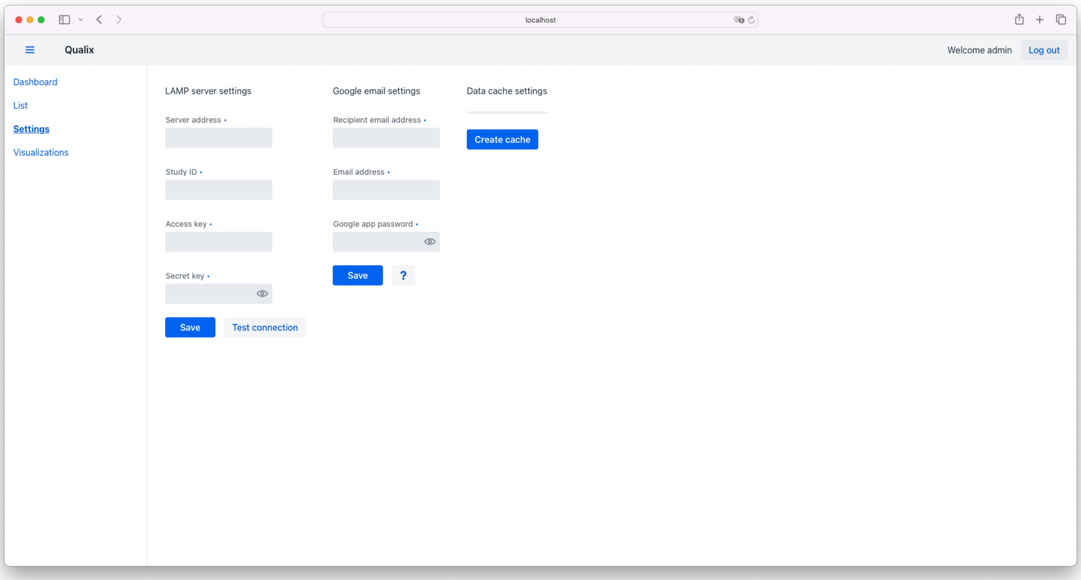
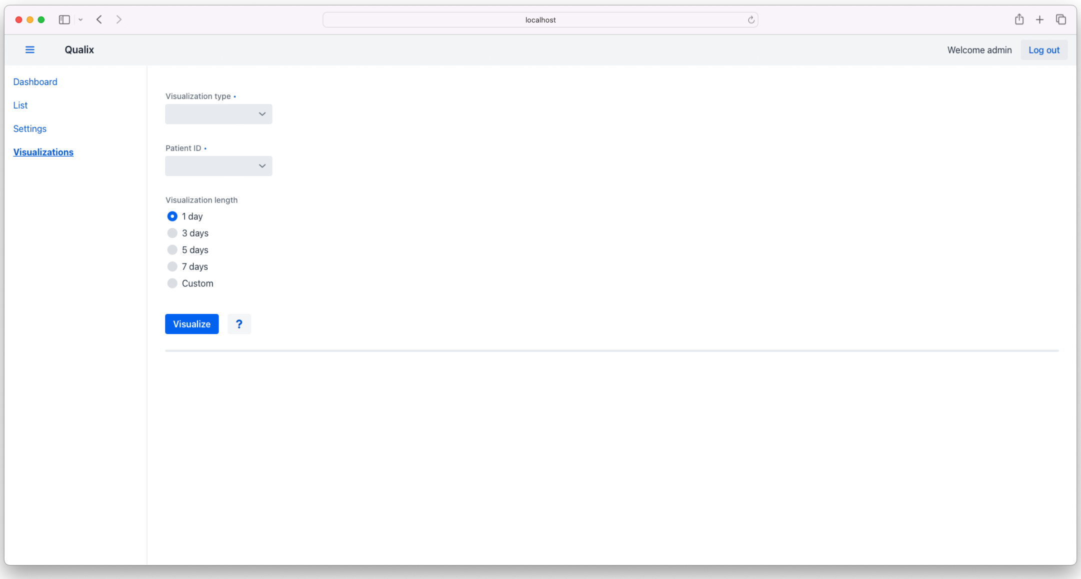
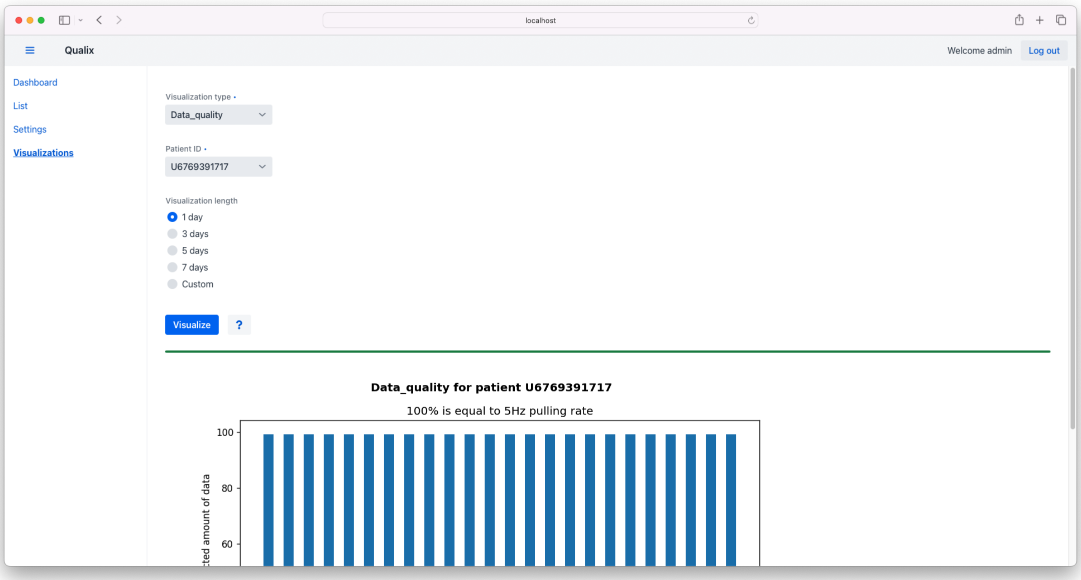


Figure 2.12. Qualix Empty Settings view.



Visualization length: [✓ 1 day]

Figure 2.13. Qualix - Empty Visualizations view.



Visualization type •
Data_quality

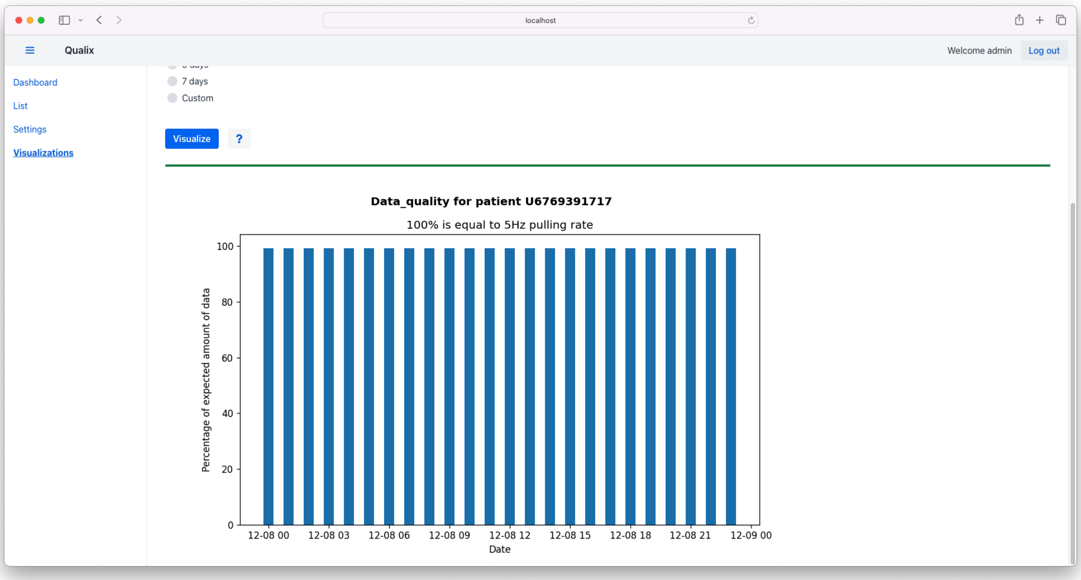
Patient ID •
U6769391717

Visualization length
[✓ 1 day]
3 days
5 days
7 days
Custom

Data_quality for patient U6769391717

100% is equal to 5Hz pulling rate

Figure 2.14. Qualix Visualizations view - Data quality visualization.



Data_quality for patient U6769391717

100% is equal to 5Hz pulling rate

Figure 2.15. Qualix Visualizations view - Full data quality visualization.

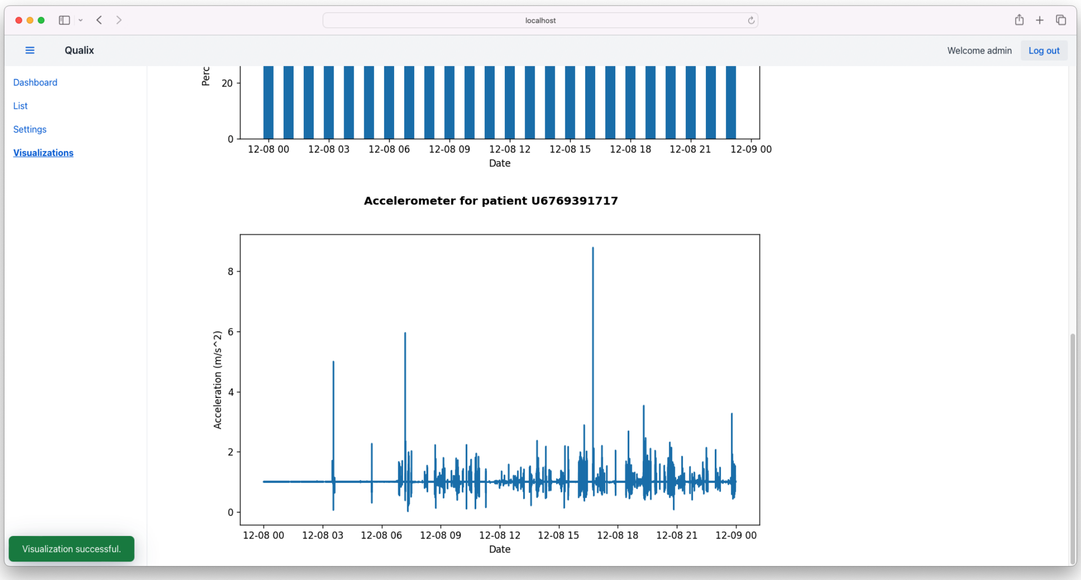


Figure 2.16. Qualix Visualizations view - Full accelerometer visualization.

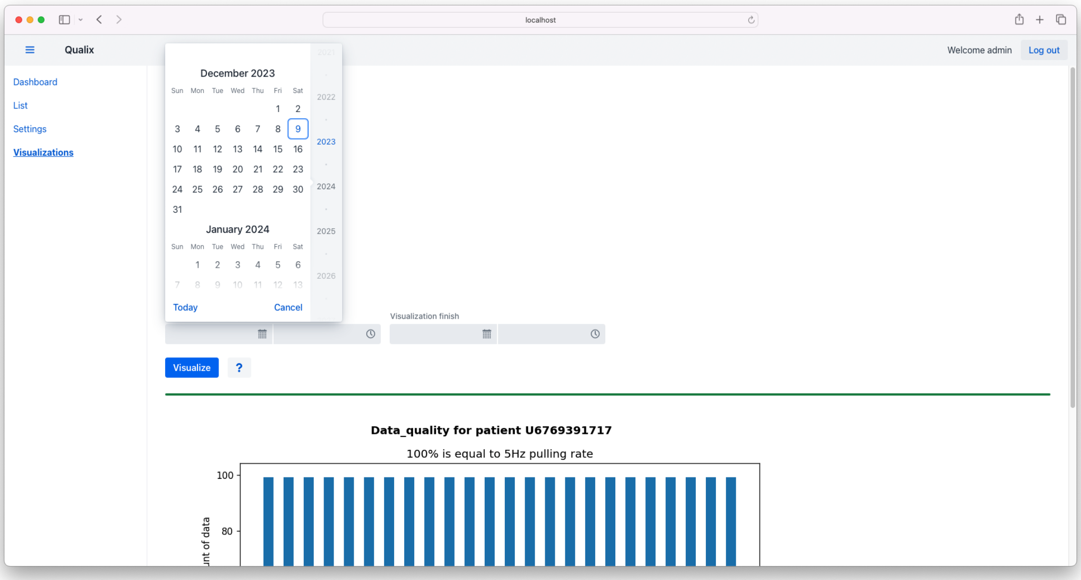
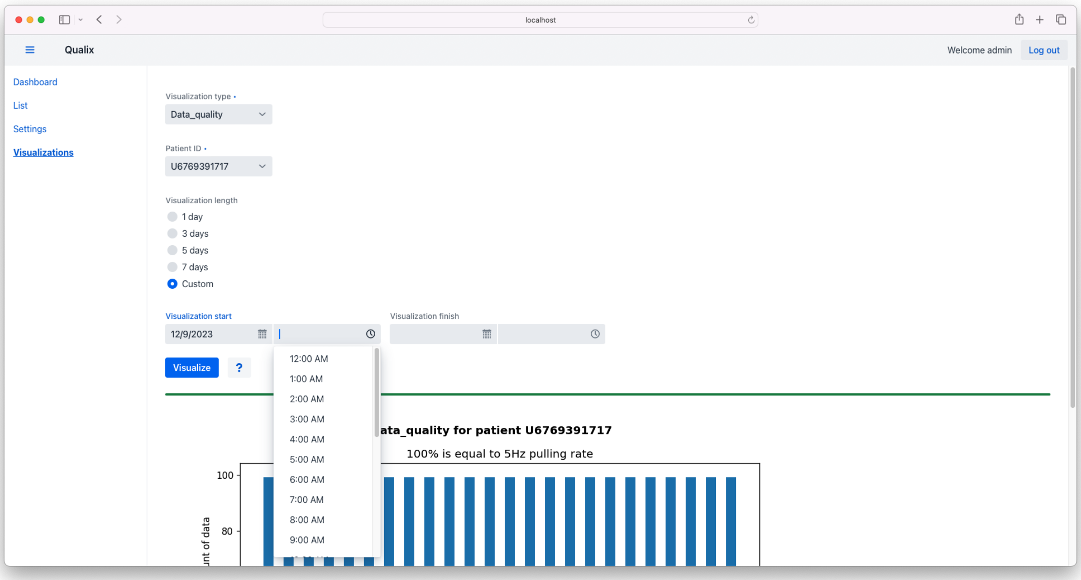


Figure 2.17. Qualix Visualizations view - Custom visualization settings - date.



Visualization type • Data_quality

Patient ID • U6769391717

Visualization length
- 1 day
- 3 days
- 5 days
- 7 days
- [✓ Custom]

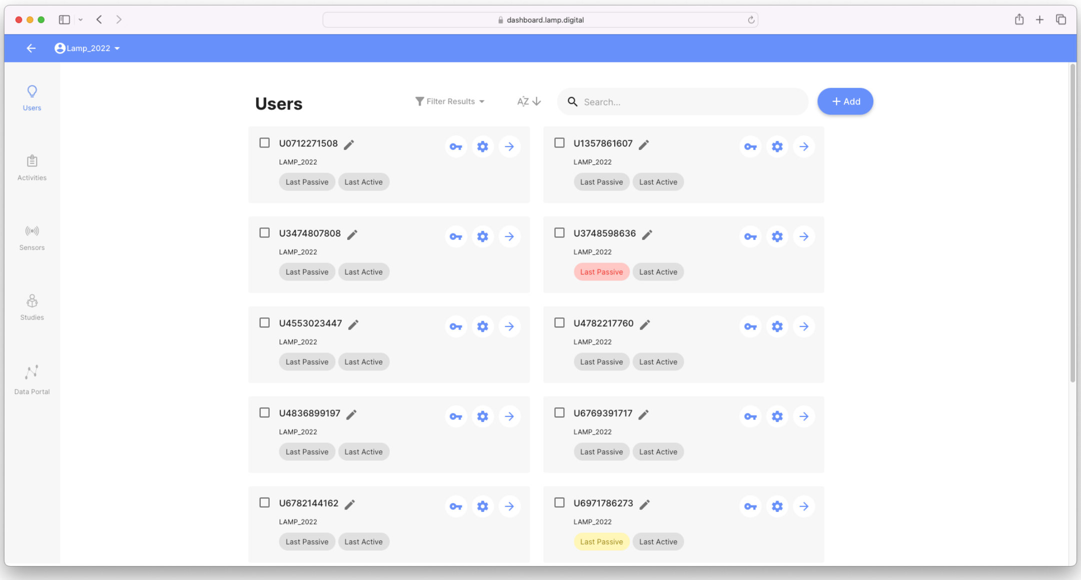
Visualization start: 12/9/2023

Visualization finish

12:00 AM
1:00 AM
2:00 AM
3:00 AM
4:00 AM
5:00 AM
6:00 AM
7:00 AM
8:00 AM
9:00 AM

ata_quality for patient U6769391717
100% is equal to 5Hz pulling rate

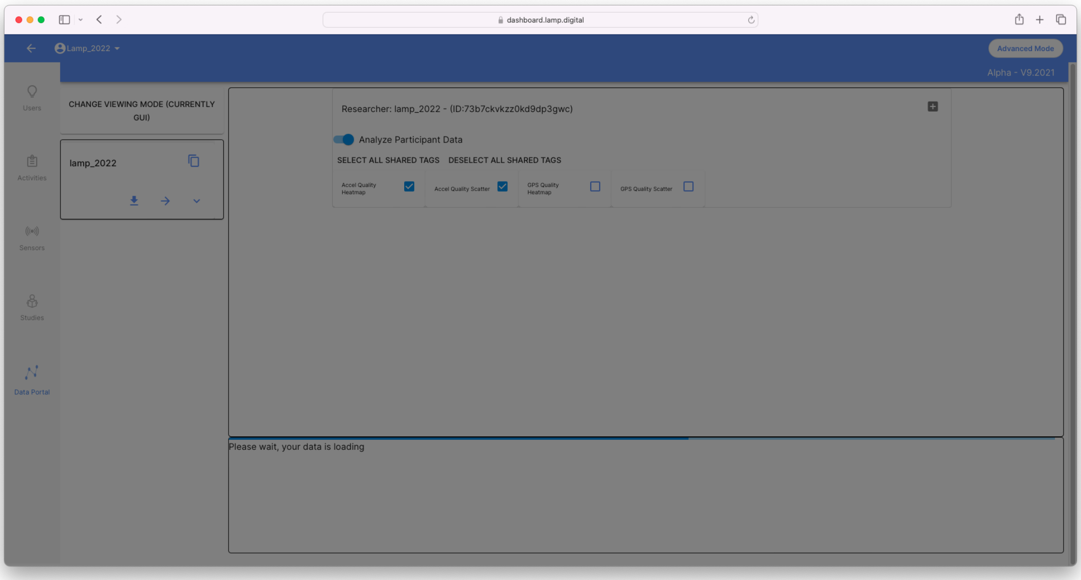
Figure 2.18. Qualix Visualizations view - Custom visualization settings - time.



Users

U0712271508 LAMP_2022 Last Passive Last Active	U1357861607 LAMP_2022 Last Passive Last Active
U3474807808 LAMP_2022 Last Passive Last Active	U3748598636 LAMP_2022 Last Passive Last Active
U4553023447 LAMP_2022 Last Passive Last Active	U4782217760 LAMP_2022 Last Passive Last Active
U4836899197 LAMP_2022 Last Passive Last Active	U6769391717 LAMP_2022 Last Passive Last Active
U6782144162 LAMP_2022 Last Passive Last Active	U6971786273 LAMP_2022 Last Passive Last Active

Figure 2.19. LAMP dashboard



Researcher: lamp_2022 - (ID:73b7ckvkzz0kd9dp3gwc)

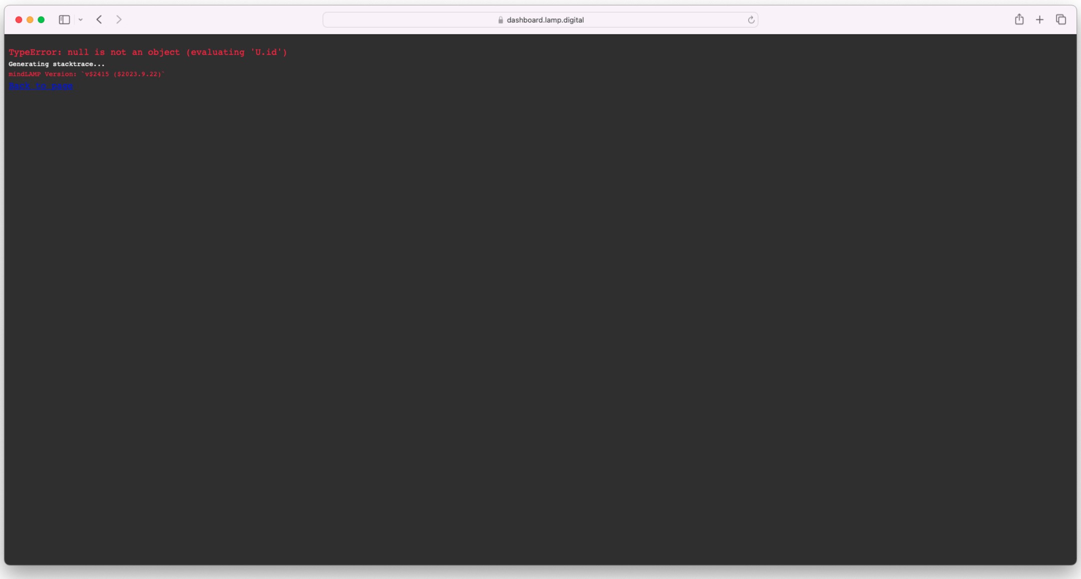
Analyze Participant Data

SELECT ALL SHARED TAGS DESELECT ALL SHARED TAGS

Accel Quality Heatmap ✓ Accel Quality Scatter ✓ GPS Quality Heatmap ☐ GPS Quality Scatter ☐

Please wait, your data is loading

Figure 2.20. Data portal not responding



```
TypeError: null is not an object (evaluating 'U.id')
Generating stacktrace...
mindLAMP Version: `v$2415 ($2023.9.22)`
Back to page
```

Figure 2.21. Data portal fails

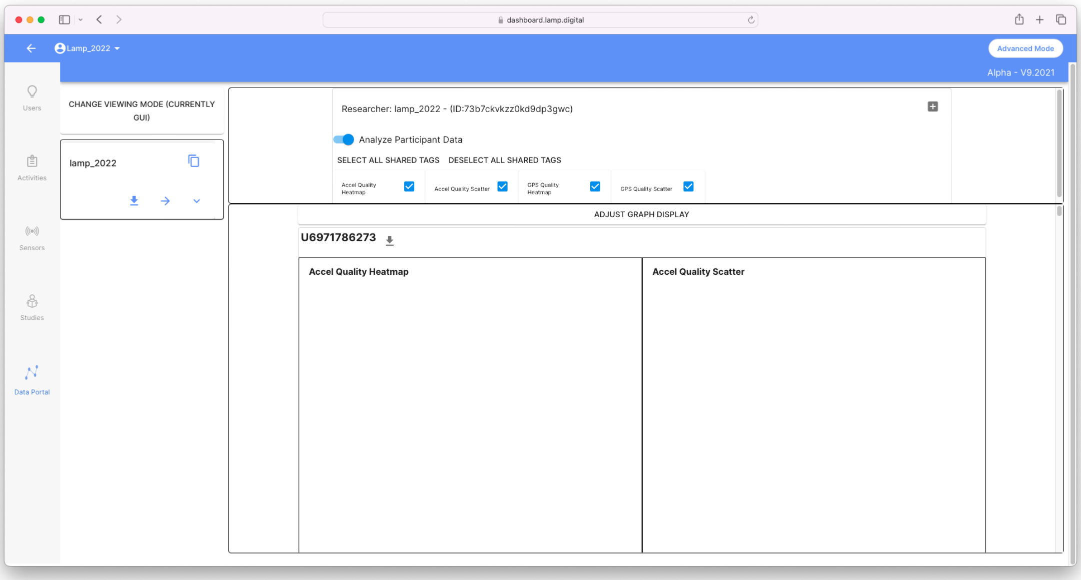


Figure 2.22. Data portal strange UI 1

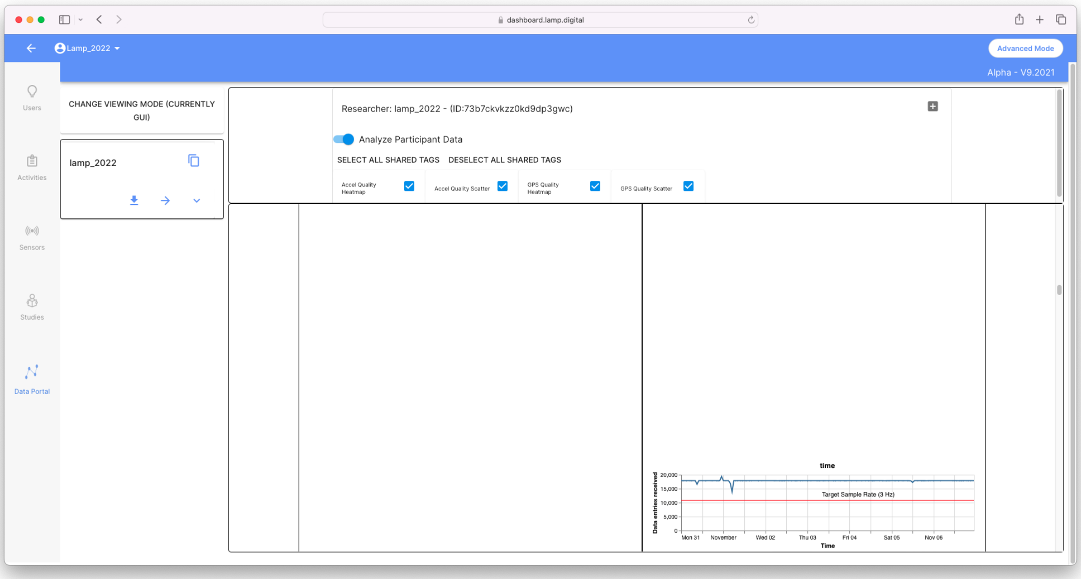


Figure 2.23. Data portal strange UI 2

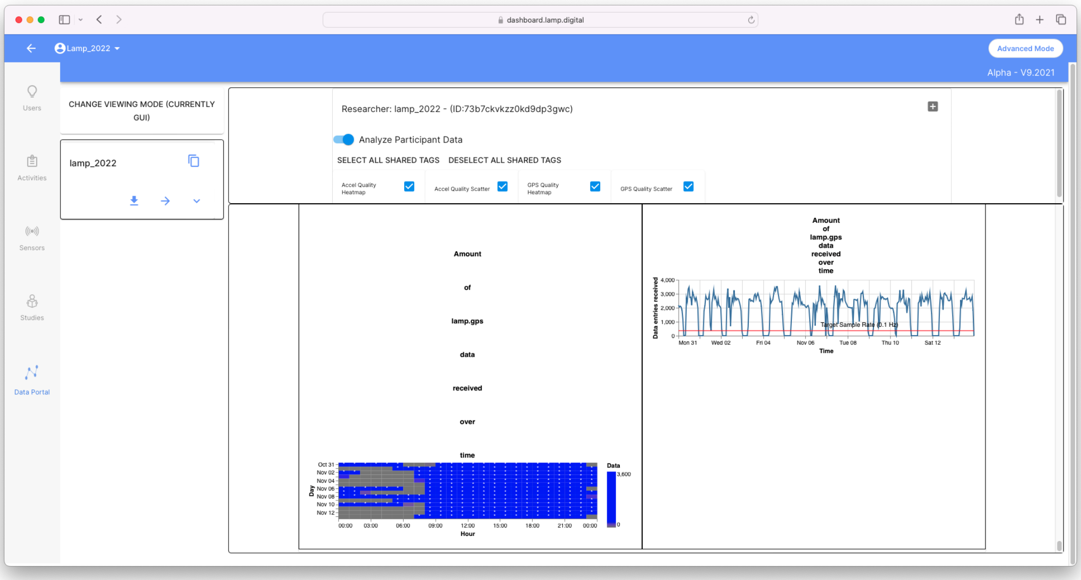


Figure 2.24. Data portal strange UI 3

Chapter 3
Weekend and Weekday Classification Using Smartphone Data

3.1 Introduction

The next part of my thesis was using the LAMP platform to gather behavioral data from smartphones and to try and distinguish between a workday and a weekend using this data. I focused on the accelerometer and screen state sensors in particular. Accelerometer can be used to measure a person's activity level [40]. It is my hypothesis, that a person's activity level changes during the weekend. People with physically demanding jobs may choose to decrease their activity level during the weekend and rest. While people with sedentary jobs may use the weekend to be outside and active as much as possible. For my purposes, it doesn't matter whether someone is more or less active during the weekend, any change from the workday will suffice. I used the same reasoning for the screen state sensor. It seems likely that a person's screen time will change during the weekend. Some people may choose to relax while using a smartphone, which may result in an increased screen time. If someone has to work with screens during the work week however, they may use the weekend for other activities, thus decreasing their screen time.

3.2 Data pre-processing and processing

I recruited 8 participants and gathered data from 26 May 2023 to 2 February 2024 which is a total of 253 days. The age range was 21-59, the average of ages 36 and the median 29.5. 2 participants were female. All of the participants agreed to have their data collected. The participants' names were anonymized using randomly generated IDs and no personal information whatsoever, including the data themselves, will be included in this thesis.

Since data from all sensors from participants' smartphones are uploaded to the LAMP server, I slightly modified the *create_data_cache.py* script described in 2.3.4 and used it to download the data in chunks. Since the accelerometer has a very high pulling rate of 5Hz, this process took a very long time. One week of data could take multiple hours to download. Since the development of Qualix was running concurrently with the study, it was unfortunately not possible to use its data cache, which would make this process much easier. Instead of taking hours to download a week of data, it would take minutes to copy the files from the Qualix data cache. After downloading, the data was stored in .csv files using the same naming structure described at 2.3.4.

For data pre-processing and processing I used MATLAB [41], since it is a software and a programming language for data processing with which I am most familiar.

3.2.1 Accelerometer

The accelerometer sensor data are sampled with a frequency of 5Hz. 3 axes are recorded, together with timestamp and UTC time. To these I added Combined acceleration a_{total}, which is given by Equation (1).

$$a_{total} = \sqrt{a_x^2 + a_y^2 + a_z^2} \tag{1}$$

Next, basic accelerometer features are calculated for each day. These features include Mean, Minimum, Maximum, Sum and Root mean square (RMS), which is defined by Equation (2).

$$x_{RMS} = \sqrt{\frac{1}{N}\sum_{n=1}^{N} |x_n|^2} \tag{2}$$

As for the Mean of Combined acceleration, the static force of gravity of 1g was subtracted. This not only makes the visualizations clearer, the reader can compare image 3.1 with gravitational pull and image 3.2 without, but this can increase the performance of the statistical model that is used to analyze the data, since the main signal that we are interested in, the user's movement, is not masked by the static gravitational pull.

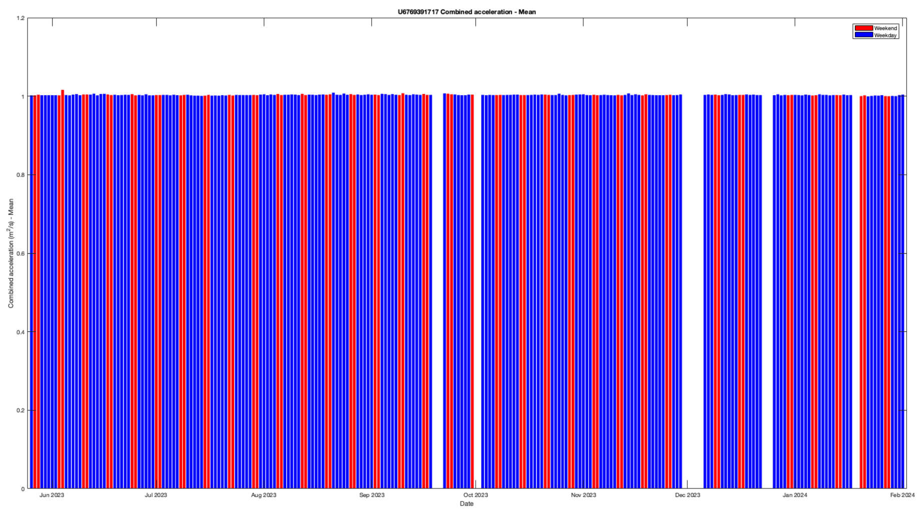


Figure 3.1. Combined Acceleration - Mean for patient U6769391717

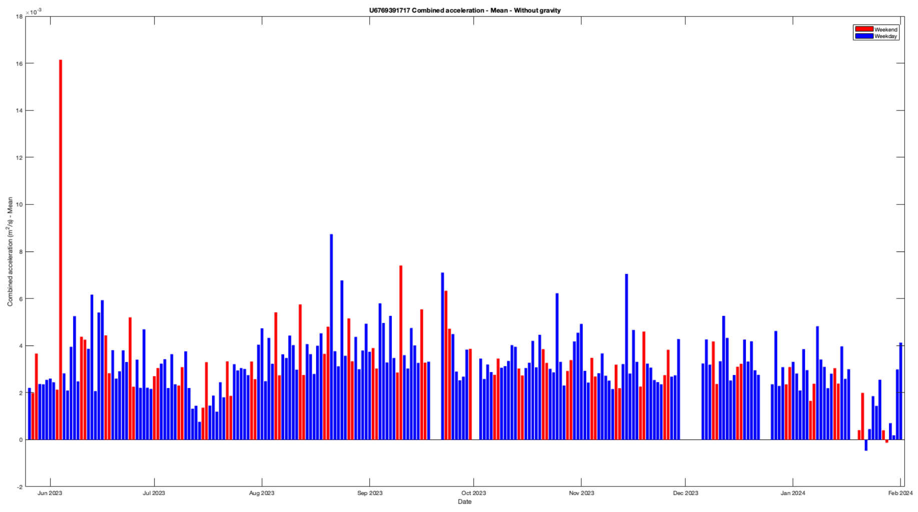


Figure 3.2. Combined Acceleration - Mean - Without gravity for patient U6769391717

After these simple features are calculated, more complex features follow. These include the least active five-hour period (L5) and the most active ten-hour period (M10) and their starting hours. L5 is the average of the sum of activities during the least active five hour period. Similarly, M10 is calculated using the most active ten hour period. A lower L5 indicates a deeper sleep with a less fragmented rhythm and a higher M10 value represents being more active during the day, when the measurement was done using an ActiGraph device [42]. But as Hekler et al. show, smartphones (in the study's case Android smartphones) can provide comparable estimates of physical activity [43]. The starting hours of M10 and L5 correspond to the first hour in which either a M10 or a L5 were found. If a participant was most active during the hours between 7:00 and 17:00 for example, the M10 starting hour would equal to 7.

Since the M10 and L5 features require that there is enough data for the whole day, days with less than 24 hours of data were excluded. This unfortunately excluded some participants from the study due to not having enough data.

3.2.2 Screen state

Similarly to the accelerometer, screen state events are stored with a timestamp and UTC time. The screen state events are represented by numbers from 0 to 3:

- 0: screen on
- 1: screen off
- 2: device locked
- 3: device unlocked

For my purposes I excluded all data points where the screen state event equaled 2 or 3. This was done because I wanted to measure how many times per day the screen lit up, not how many times the phone was locked/unlocked. Apart from that, I calculated the total screen time in minutes per day. To do that, I started by separating the data into daily chunks and then removed the first data point if it didn't contain 0 and the last data point if it didn't contain 1. Next, it is logical that the rest of the rows should follow a simple pattern of repeating 0s and 1s. Since after every event where the is screen turned on, the next event should be for it to turn off. Same thing applies for an event where the screen is turned off, since it is not possible to turn off the screen when it's already off. There were some rows of repeating 1s or 0s however and these needed to be removed. For the rows of repeating 0s I always kept the last row, since this repeating pattern could only happen if there are some missing rows where screen state is equal to 1. Similarly with rows of repeating 1s I always kept the first row, since some rows with 0s were missing. This could change the result of the calculation, but fortunately these events were quite rare. After this pre-processing a simple algorithm goes through the table and calculates the total time of minutes used for each day:

```
for j = 1:2:height(daily_chunk)
    minutes_used = minutes_used + minutes(daily_chunk.UTC_time(j+1)
    - daily_chunk.UTC_time(j));
end
```

3.3 SVM

For classifying the data as either a weekend or a weekday, I chose to use the support vector machine (SVM) algorithm. This algorithm is a supervised machine learning model used to classify data. SVM tries to find a hyperplane that separates the data points into classes, in my case into either a Weekend class or a Weekday class. The data can either be linearly separated in the input space, or in a higher dimension feature space. Kernel functions are used to transform the data into a higher dimensional space where a hyperplane can separate the data easily. [44]

In the preliminary testing, many SVMs with different kernels were trained, including Linear, Quadratic, Cubic and Gaussian. For the Gaussian kernel, different kernel scales were used. A smaller kernel scale will result in a more complex boundary, which can lead to training data overfitting. A larger kernel scale on the other hand will result in a less complex, more general boundary, which can cause training data underfitting. The best results were obtained with a Gaussian kernel with a medium kernel scale of $\sqrt{P}$, where P is equal to the number of predictors. All results shown are therefore achieved with a Gaussian kernel with a medium kernel scale.

In total I trained three different SVMs for each participant. For the first one, only features derived from the accelerometer sensor were used. For the second one, only features derived from the screen state were used and for the last one, features from both sensors were used. This was done because I wanted to see how the results compared with each other and which model produced the best result. A table 3.1 shows which features were derived for each sensor.

Accelerometer	Screen state
Combined acceleration - Mean	Minutes used
Combined acceleration - Min	Times turned on
Combined acceleration - Max	
Combined acceleration - Sum	
Combined acceleration - RMS	
M10	
M10 - Start hour	
L5	
L5 - Start hour	

Table 3.1. Features used to train SVM by sensor.

Since there are five weekdays in a week, the misclassification costs had to be altered, otherwise the models would classify each day as a weekday and achieve an accuracy of $7/5 \approx 0.71$. To mitigate this, the cost of misclassifying a weekend as a weekday was multiplied by a factor of 2.5. In order to evaluate the models I used a holdout of 10% and in order to prevent overfitting a 10-fold cross-validation was used.

3.3.1 Results

The accuracies of the SVMs on unseen data are found in table 3.2. Only four participants out of eight are included in the table, since the rest had insufficient amount of data to train the models. The models showed a wide range of accuracies, from high levels above 0.70 for participant U3748598636, to low levels around 0.50 for participant U6971786273.

Patient ID	Accelerometer	Screen state	Both sensors	Days of data
U6769391717	0.74	0.70	0.61	236
U3748598636	0.76	0.79	0.71	147
U4782217760	0.60	0.60	0.80	50
U6971786273	0.43	0.57	0.50	140

Table 3.2. SVMs accuracies on unseen data.

This disparity in the performance between participants can be attributed to several factors, such as data quality and the behavioral patterns of participants. A detailed analysis and interpretation of these results is discussed in 4.2.2.

Chapter 4
Discussion

4.1 Qualix application

4.1.1 Summary

The Qualix application was developed as an add-on application to the LAMP platform. The goal was to service the needs for automatic data quality check, data quality notifications and easily generated visualizations, since these features are missing from the LAMP platform. From this perspective, the application fulfilled its goals, since all of the aforementioned features were implemented and are fully functional.

Another way to address the limitations of the LAMP platform would be to fix them directly within the platform itself, since its code is open source. I decided not to pursue this route however, since the code base is very large and the communication with the LAMP team would likely hinder my progress. Also, with the freedom that comes from developing a bespoke application from scratch, I was able to meet the needs of my supervisor and his team for how the Qualix application should look and behave. This level of customization would likely not be possible just by modifying the LAMP platform itself, since the changes would have to be drastic and would likely be rejected by the LAMP team. It is also uncertain whether these changes would be actually welcomed. For example the CORTEX package for data quality and visualization was designed to work exactly as it does. The reason that I and my colleagues find it frustrating to use, doesn't mean that the LAMP team didn't have a valid reason for implementing it this way. The same holds true for the LAMP Dashboard — image 2.19. Just because I find it cumbersome to use doesn't mean that someone else wouldn't find it easy to use and well-arranged.

Qualix is easy to build and deploy, both locally and on the cloud, thanks to the use of Docker containers and the included build and deploy script. The goal was to make the installation as easy as possible, compared to the installation of the LAMP platform itself. I am confident in saying that whoever is able to deploy the LAMP platform will have no problems deploying Qualix. One of the reasons for this ease of use is the fact that Qualix is pre-configured with default values and settings, such as usernames and passwords, database and port settings. This means that Qualix will work out of the box just by running the included build and deploy script, without the need for any configuration whatsoever. This approach has a downside as well however, which brings us to limitations.

4.1.2 Limitations and future work

The main limitation comes in the form of less than ideal security. Right now Qualix user authentication is implemented using a basic `InMemoryUserDetailsManager`, which as the name suggests, stores the usernames and passwords in memory. Not only that, the usernames and passwords are stored directly in the *SecurityConfig.java* file. This is not ideal and the authentication provider should be changed in the future to something else, such as to the lightweight directory access protocol (LDAP).

As for the visualization, currently only 2 visualizations are supported: Accelerometer and Data quality. These are the most important ones in my opinion, especially the data quality visualization. The data quality visualization also uses the accelerometer sensor, since this sensor is always available and doesn't require any sort of external signal to function. This makes it suitable for measuring data quality, since even when the GPS or internet signals become unavailable, the accelerometer data can still be cached and sent to the LAMP platform once the device is reconnected to the internet. Using other sensors in the visualization could be beneficial however, since using only the accelerometer doesn't paint the whole picture regarding collected data quality. Visualizations for other sensors could be added in the future to increase the functionality of Qualix.

The dashboard view could also be refined, specifically the data quality calculation mechanism. Currently the last time a row of data was received is used to determine data quality. It is theoretically possible, even though unlikely, that a device could send only one row of data within the 2 hour window when the data quality check is done. This could then mislead the study staff into thinking that there are plenty of data from such a participant, even though the opposite would be the case. This would be detected by checking the data quality visualization however. Still, it is a suitable area for future development.

The last area of future development could be adding a new view for downloading participants' data. Since the data cache is already generated, it could be used to download participants' data from all sensors for the last week almost instantly. Right now, using the official documented way of downloading behavioral data is with the LAMP python library. This process can take a very long time, since the documentation recommends downloading the data in chunks of one thousand rows. This means that for every one thousand rows of data a request to the LAMP server has to be made. If we take for example the accelerometer sensor with a 5Hz pulling rate, this means that a request to the LAMP server has to be made to download only two hundred seconds worth of accelerometer data for one participant. This request obviously has to be processed by the server, the required thousand rows have to be extracted from the database, then sent to the client, where they are saved into memory and later to a data file. This process creates a lot of overhead which slows down the saving operation. The user can choose to ignore the recommended thousand row limit, this can however overwhelm the LAMP server very quickly which slows the download operation even more. Right now, using Qualix, the user can manually copy the entire data cache and "download" a week worth of data for all participants in seconds. This is quicker than using the LAMP python library, but it is not very user friendly, since the user has to have access to the server where Qualix is deployed and access the data cache inside the Docker container. So a new view for downloading user data would be useful. It is important to also note that right now, if users want to download the data using the LAMP python library, they have to have the login information in order to connect to the LAMP server. This is not always desirable, since the study administrator might want to give access to the anonymized user data to the study staff without giving them the access to the whole LAMP server. This would be possible using a new view for downloading data in Qualix, which could be accessible to ordinary users, not only to study administrators.

4.2 Weekend/Weekday Classification Study

4.2.1 Data collection

I've encountered some challenges during the data collection phase. These were mainly centered around data quality, especially for users with Android devices and around high battery drain. Since there are numerous manufactures of Android devices, there are also numerous manufacturers of the sensors inside these smartphones. These sensors can be of varying quality and can have different capacities for providing data. For example the accelerometer sensor on iPhones continuously provided a pulling rate of 5Hz. The same can not be said of the Android devices in our study. It is even difficult to quantify the exact pulling rate of their accelerometers, because it would change constantly and have spots of missing data. This could be because of the accelerometer sensor itself. It may have some capability by design, let's say 1Hz for example and the operating system would bombard it with a request for data five times per second. Apart from this, other applications or even the operating system itself would likely need the data for themselves sometime, which could overload the sensor.

The next problem could reside in the operating systems themselves. Since there are multiple versions of Android[1] from multiple manufacturers, the data collection applications themselves have to support them, which leads to a more general rather than more specific code design. It is much easier to design an application that focuses on a handful of devices by the same manufacturer than it is to create that same application that will support hundreds of devices from numerous manufacturers.

It is the constant access to smartphone sensors, especially the GPS and accelerometer that causes the high battery drain that we've seen during the study. It was due to this high battery drain that two participants dropped out. In the study the extra battery drain was in the range of around 30% per day. This causes increased wear on the battery and can become bothersome, especially for users with older devices, worn out batteries or for users that need their phones to last as long as possible, such as when on a camping trip.

There are ways to mitigate these problems however. For the poor data quality, giving the data collection app necessary permissions[2], not using low power/battery saving mode and opening the data collection application from time to time, since the operating system may revoke access to sensors to applications that have not been opened for some time or even terminate the application entirely. For these solutions to work however, the study staff have to know that a data quality problem has arisen. Qualix can help with this, since a member of the study staff can receive an email notification and see that the LAMP platform didn't receive data from some participants. These participants can then be contacted and asked to open the data collection application, not use low power mode, etc.

As for the high battery drain, one solution could be to lower the pulling rate of the sensors, this has to be done with caution however, since a pulling rate that's too low can cause important data points to be missed, which can negatively impact clinical

[1] There are multiple versions of iOS as well, but Apple allows even very old iPhones access to their newest operating system and installs updates automatically by default, so more users are using the same (the latest) version of iOS.

[2] It is also important to note that when designing a study, it is much more simple for the study staff to create a tutorial showing how to grant these permissions on iOS than it is on Android, since different versions of Android from different manufacturers can have different settings that need to be changed, while different iOS versions look and behave almost the same.

interpretations [33]. Another solution is to offload the data collection onto another device. The participants can be given another smartphone, which would be used only for data collection, or they could offload certain sensors, such as the accelerometer by using a wearable device, such as one of the ActiGraph devices[3]. This creates another issues however. In the case of the extra smartphone, the participant doesn't have to worry about battery running out on his personal smartphone, but they now have to keep another device charged and on their person. The data collected may not be perfect either, since they will not use the new smartphone in the same way that they use their personal one. As for using wearable devices, this can be a better option and it only becomes an issue for some users that don't feel comfortable wearing a watch-like device. It is also easier to forget to wear an ActiGraph device than it is to forget to take your smartphone with you.

4.2.2 Results

The SVMs classification accuracies are of a wide range. For some participants, such as for U3748598636, the models performed well across both sensors. This indicates that the participant behaves differently during the weekend than they do during the week and that this behavior change was recorded into the way they used their phone.

For participant U6971786273 the models didn't detect any change in behavior during the weekend. This could be due to a number of reasons. One of them is that the participant simply doesn't change their phone use that much during the weekend compared to weekdays or they change it in such a way that it is not detectable by the SVM. It is possible that the features that I selected weren't specific enough for this purpose or there weren't enough of them. Another issue could be with only using the accelerometer and screen state sensors and not using the GPS sensor. I've made the decision to omit it, since the accelerometer and screen state sensor are always available. Even in a place where there is no internet connection, the data collection app will cache data from these sensors and then upload them to the LAMP server once an internet connection is established. The same is not true for GPS, since it needs a clear view of the sky to work and struggles in densely populated areas. These issues would lead to missing data or incorrect data which could skirt the results, but since the accelerometer and screen state sensors didn't produce satisfactory results, the GPS sensor is a good avenue for future research.

As for the participants that didn't have enough data, two of them dropped out shortly due to high battery drain caused by the data collection application and the other two were using Android devices. As described in 4.2.1, Android devices didn't perform well for various reasons.

Since Qualix was developed concurrently with the weekend/weekday classification study, it was not possible to use it to check the data quality and to notify the study staff (in this case me) via email. It would be interesting to see, how much would the data quality improve if Qualix was used. Due to time constraints, it was unfortunately not possible to do a second round of the study, but it is an interesting avenue for future research.

[3] `https://theactigraph.com/wearable-devices`

Chapter 5
Conclusion

Behavioral data are ever-present in our modern world. They can be used to infer information about us, such as our mood or extravertedness/introvertedness. This information can then be used to influence our behavior, both positively and negatively, voluntarily and involuntarily.

There are open source platforms used for the collection of behavioral data. One of them is the LAMP platform. During our testing, we've found some limitations of this platform. That is why I've developed a full-stack web application called Qualix which connects to the LAMP platform and adds missing features, such as email notifications, data quality checks and fast visualizations.

Using the LAMP platform, I've collected behavioral data from 8 participants for 253 days. For some participants, the results are promising and show that behavioral data collected through smartphones can be used to detect behavioral patterns that can classify days into weekends and weekdays. There is room for improvement however, as for some participants the classification model had a low accuracy and some participants didn't have enough data and had to be excluded. An improved data quality checking, such as through the Qualix application, better feature selection and model optimization could be used to enhance the performce of the classification models.

References

[1] Merriam-Webster. *Behavior*.
`https://www.merriam-webster.com/dictionary/behavior`.

[2] Alex Gomez-Marin, Joseph J Paton, Adam R Kampff, Rui M Costa, and Zachary F Mainen. Big behavioral data: psychology, ethology and the foundations of neuroscience. *Nature Neuroscience*. 2014, 17 (11), 1455–1462. DOI 10.1038/nn.3812.

[3] Gabriella M. Harari, Nicholas D. Lane, Rui Wang, Benjamin S. Crosier, Andrew T. Campbell, and Samuel D. Gosling. Using Smartphones to Collect Behavioral Data in Psychological Science: Opportunities, Practical Considerations, and Challenges. *Perspectives on Psychological Science*. 2016, 11 (6), 838–854. DOI 10.1177/1745691616650285.

[4] Fani Tsapeli, and Mirco Musolesi. Investigating causality in human behavior from smartphone sensor data: a quasi-experimental approach. *EPJ Data Science*. 2015, 4 (1). DOI 10.1140/epjds/s13688-015-0061-1.

[5] Rui Wang, Fanglin Chen, Zhenyu Chen, Tianxing Li, Gabriella Harari, Stefanie Tignor, Xia Zhou, Dror Ben-Zeev, and Andrew T. Campbell. *StudentLife: assessing mental health, academic performance and behavioral trends of college students using smartphones*. In: *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 2014.

[6] Sohrab Saeb, Mi Zhang, Christopher J Karr, Stephen M Schueller, Marya E Corden, Konrad P Kording, and David C Mohr. Mobile Phone Sensor Correlates of Depressive Symptom Severity in Daily-Life Behavior: An Exploratory Study. *Journal of Medical Internet Research*. 2015, 17 (7), e175. DOI 10.2196/jmir.4273.

[7] William Douglas Evans, Lorien C. Abroms, David Broniatowski, Melissa A. Napolitano, Jeanie Arnold, Megumi Ichimiya, and Sohail Agha. Digital Media for Behavior Change: Review of an Emerging Field of Study. *International Journal of Environmental Research and Public Health*. 2022, 19 (15), 9129. DOI 10.3390/ijerph19159129.

[8] Melissa A Napolitano, Jessica A Whiteley, Meghan Mavredes, Ashley Hogan Tjaden, Samuel Simmens, Laura L Hayman, Jamie Faro, Ginger Winston, Steven Malin, and Loretta DiPietro. Effect of tailoring on weight loss among young adults receiving digital interventions: an 18 month randomized controlled trial. *Translational Behavioral Medicine*. 2021, 11 (4), 970–980. DOI 10.1093/tbm/ibab017.

[9] Melissa A. Napolitano, Jessica A. Whiteley, Meghan N. Mavredes, Jamie Faro, Loretta DiPietro, Laura L. Hayman, Charles J. Neighbors, and Samuel Simmens. Using social media to deliver weight loss programming to young adults: Design and rationale for the Healthy Body Healthy U (HBHU) trial. *Contemporary Clinical Trials*. 2017, 60 1–13. DOI 10.1016/j.cct.2017.06.007.

[10] W. Douglas Evans, Jeffrey B. Bingenheimer, Michael Long, Khadidiatou Ndiaye, Dante Donati, Nandan M. Rao, Selinam Akaba, Ifeanyi Nsofor, and Sohail Agha.

Outcomes of a social media campaign to promote COVID-19 vaccination in Nigeria. *PLOS ONE*. 2023, 18 (9), e0290757. DOI 10.1371/journal.pone.0290757.

[11] Erin E. Bonar, José A. Bauermeister, Frederic C. Blow, Amy S.B. Bohnert, Carrie Bourque, Lara N. Coughlin, Alan K. Davis, Autumn Rae Florimbio, Jason E. Goldstick, Diane M. Wisnieski, Sean D. Young, and Maureen A. Walton. A randomized controlled trial of social media interventions for risky drinking among adolescents and emerging adults. *Drug and Alcohol Dependence*. 2022, 237 109532. DOI 10.1016/j.drugalcdep.2022.109532.

[12] Erin E. Bonar, Jason E. Goldstick, Lyndsay Chapman, José A. Bauermeister, Sean D. Young, Jenna McAfee, and Maureen A. Walton. A social media intervention for cannabis use among emerging adults: Randomized controlled trial. *Drug and Alcohol Dependence*. 2022, 232 109345. DOI 10.1016/j.drugalcdep.2022.109345.

[13] Cambridge Dictionary. *Surveillance capitalism*. `https://dictionary.cambridge.org/dictionary/english/surveillance-capitalism`.

[14] Christian Montag, Bernd Lachmann, Marc Herrlich, and Katharina Zweig. Addictive Features of Social Media/Messenger Platforms and Freemium Games against the Background of Psychological and Economic Theories. *International Journal of Environmental Research and Public Health*. 2019, 16 (14), 2612. DOI 10.3390/ijerph16142612.

[15] Adam Alter. *Irresistible*. New York: Penguin Press, 2017. ISBN 9781594206641. Includes bibliographical references (pages 323-344) and index.

[16] Aarif Alutaybi, John McAlaney, Angelos Stefanidis, Keith Phalp, and Raian Ali. *Designing Social Networks to Combat Fear of Missing Out*. In: *Electronic Workshops in Computing*. BCS Learning and Development, 2018.

[17] B. F. Skinner. A case history in scientific method. *American Psychologist*. 1956, 11 (5), 221–233. DOI 10.1037/h0047662.

[18] Christopher D. Fiorillo, Philippe N. Tobler, and Wolfram Schultz. Discrete Coding of Reward Probability and Uncertainty by Dopamine Neurons. *Science*. 2003, 299 (5614), 1898–1902. DOI 10.1126/science.1077349.

[19] Chiang Ting, Xiaodan Liu, Ruoxi Lin, Guangqi Shi, and Zihan Yang. The Intrigue of Intermittent Reward and Cost: Unveiling Their Influence in Real Life. *Interdisciplinary Humanities and Communication Studies*. 2024, 1 (4). DOI 10.61173/xgzr8f61.

[20] Björn Lindström, Martin Bellander, David T. Schultner, Allen Chang, Philippe N. Tobler, and David M. Amodio. A computational reward learning account of social media engagement. *Nature Communications*. 2021, 12 (1). DOI 10.1038/s41467-020-19607-x.

[21] Brahim Zarouali, Tom Dobber, Guy De Pauw, and Claes de Vreese. Using a Personality-Profiling Algorithm to Investigate Political Microtargeting: Assessing the Persuasion Effects of Personality-Tailored Ads on Social Media. *Communication Research*. 2020, 49 (8), 1066–1091. DOI 10.1177/0093650220961965.

[22] Hannes Grassegger, and Mikael Krogerus. The data that turned the world upside down. *Vice Motherboard*. 2017, 28

[23] Carole Cadwalladr. Facebook suspends data firm hired by Vote Leave over alleged Cambridge Analytica ties. *The Guardian*.

[24] Stephanie Kirchgaessner. Cambridge Analytica used data from Facebook and Politico to help Trump. *The Guardian*. 2017, 26

[25] Nicholas Confessore. Cambridge Analytica and Facebook: The scandal and the fallout so far. *The New York Times*. 2018, 4

[26] Dipayan Ghosh, and Ben Scott. Facebook's new controversy shows how easily online political ads can manipulate you. *Time*.

[27] Tiffany Hsu. For many Facebook users, a 'last straw'that led them to quit. *The New York Times*. 2018, 21

[28] Meta. *Q4 2019 Earnings Report*. Online. 2019. `https://s21.q4cdn.com/399680738/files/doc_financials/2019/q4/Q4-2019-Earnings-Presentation-_final.pdf`. Slide 2.

[29] Kristen Herhold. How people view Facebook after the Cambridge Analytica data breach. *The Manifest*.

[30] Joanne Hinds, Emma J. Williams, and Adam N. Joinson. "It wouldn't happen to me": Privacy concerns and perspectives following the Cambridge Analytica scandal. *International Journal of Human-Computer Studies*. 2020, 143 102498. DOI 10.1016/j.ijhcs.2020.102498.

[31] Yuqiong Zhu, Haihan Chen, Junda Li, Xian Mei, and Wenjuan Wang. Effects of different interventions on internet addiction: a systematic review and network meta-analysis. *BMC Psychiatry*. 2023, 23 (1). DOI 10.1186/s12888-023-05400-9.

[32] Shahana Ayub, Lakshit Jain, Shanli Parnia, Anil Bachu, Rabeea Farhan, Harendra Kumar, Amanda Sullivan, and Saeed Ahmed. Treatment Modalities for Internet Addiction in Children and Adolescents: A Systematic Review of Randomized Controlled Trials (RCTs). *Journal of Clinical Medicine*. 2023, 12 (9), 3345. DOI 10.3390/jcm12093345.

[33] Danielle Currey, and John Torous. Increasing the value of digital phenotyping through reducing missingness: a retrospective review and analysis of prior studies. *BMJ Mental Health*. 2023, 26 (1), e300718. DOI 10.1136/bmjment-2023-300718.

[34] John Torous, Hannah Wisniewski, Bruce Bird, Elizabeth Carpenter, Gary David, Eduardo Elejalde, Dan Fulford, Synthia Guimond, Ryan Hays, Philip Henson, Liza Hoffman, Chun Lim, Michael Menon, Valerie Noel, John Pearson, Randy Peterson, Ammu Susheela, Haley Troy, Aditya Vaidyam, Emma Weizenbaum, John A. Naslund, and Matcheri Keshavan. Creating a Digital Health Smartphone App and Digital Phenotyping Platform for Mental Health and Diverse Healthcare Needs: an Interdisciplinary and Collaborative Approach. *Journal of Technology in Behavioral Science*. 2019, 4 (2), 73–85. DOI 10.1007/s41347-019-00095-w.

[35] Aditya Vaidyam, John Halamka, and John Torous. Enabling Research and Clinical Use of Patient-Generated Health Data (the mindLAMP Platform): Digital Phenotyping Study. *JMIR mHealth and uHealth*. 2022, 10 (1), e30557. DOI 10.2196/30557.

[36] Rebecca Bilden, Dcurrey88, Aditya Vaidyam, Suraj Patel, Ashley Meyer, Luke Scheuer, Ertjlane, Tanvi Lakhtakia, Lucy Gray, Ryan D'Mello, Carlan1, Ryan Hays, Jtorous, Sarah Chang, Mauro Curbelo, Hunter Lester, and Toujames. *BIDMCDigitalPsychiatry/LAMP-platform: Release 2023.2.15*. 2023.

[37] Lukáš Sláma. *Behavioral and actigraphic data processing and visualization*. 2021. `http://hdl.handle.net/10467/96671`. Accessed: 23-04-2024.

[38] In: *Learn Objective-C for Java Developers*. Apress, 2009. 353–402. ISBN 9781430223702.

[39] J. D. Hunter. Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*. 2007, 9 (3), 90–95. DOI 10.1109/MCSE.2007.55.

[40] Yuyang Zhai, Navina Nasseri, Jana Pöttgen, Eghbal Gezhelbash, Christoph Heesen, and Jan-Patrick Stellmann. Smartphone Accelerometry: A Smart and Reliable Measurement of Real-Life Physical Activity in Multiple Sclerosis and Healthy Individuals. *Frontiers in Neurology*. 2020, 11 DOI 10.3389/fneur.2020.00688.

[41] The MathWorks Inc.. *24.1.0.2578822 (R2024a) Update 2*. 2024. `https://www.mathworks.com`.

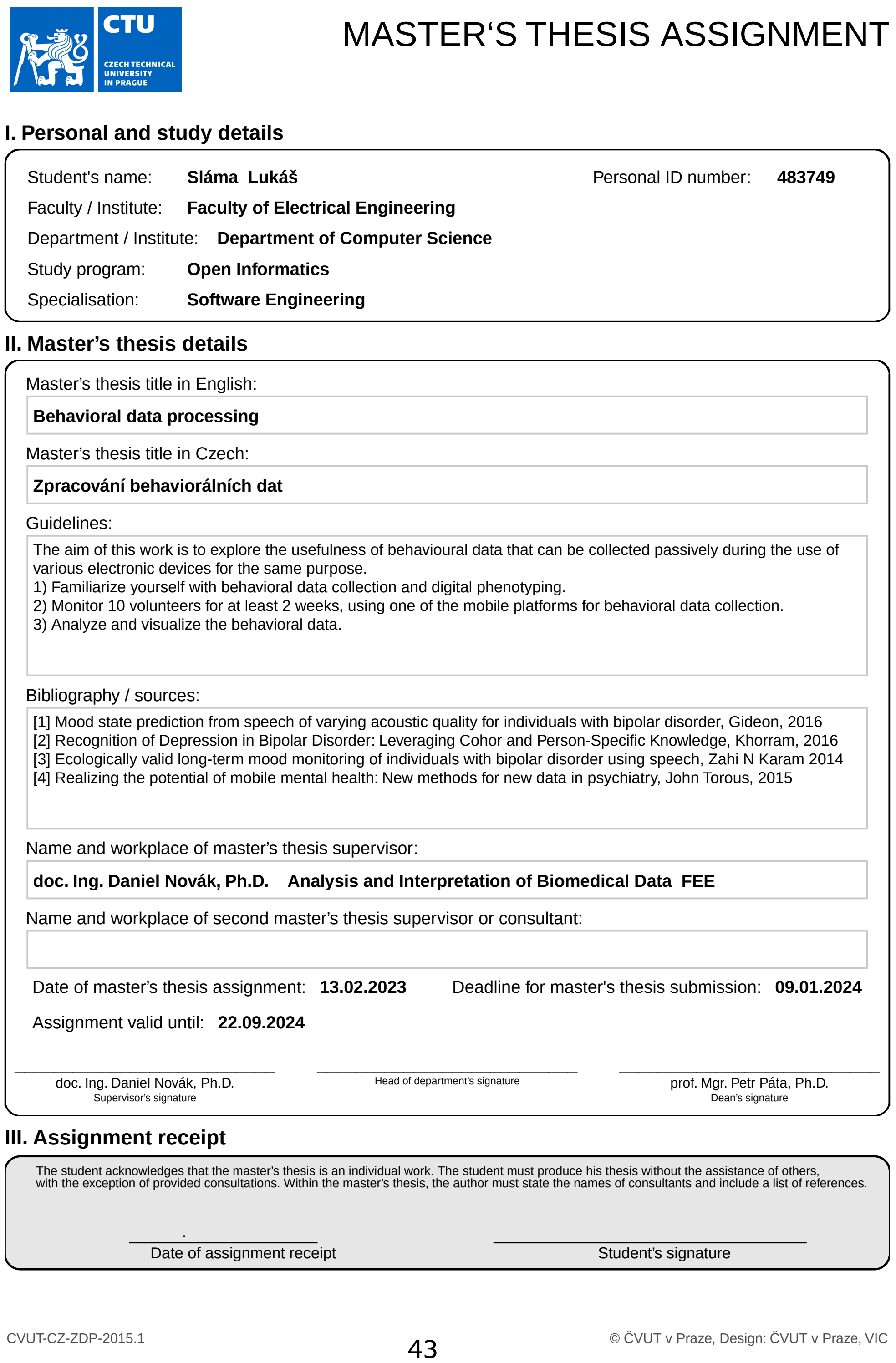
[42] BSB Gonçalves, Taísa Adamowicz, Fernando Mazzilli Louzada, Claudia Roberta Moreno, and John Fontenele Araujo. A fresh look at the use of nonparametric analysis in actimetry. *Sleep Medicine Reviews*. 2015, 20 84–91. DOI 10.1016/j.smrv.2014.06.002.

[43] Eric B Hekler, Matthew P Buman, Lauren Grieco, Mary Rosenberger, Sandra J Winter, William Haskell, and Abby C King. Validation of Physical Activity Tracking via Android Smartphones Compared to ActiGraph Accelerometer: Laboratory-Based and Free-Living Validation Studies. *JMIR mHealth and uHealth*. 2015, 3 (2), e36. DOI 10.2196/mhealth.3505.

[44] M.A. Hearst, S.T. Dumais, E. Osuna, J. Platt, and B. Scholkopf. Support vector machines. *IEEE Intelligent Systems and their Applications*. 1998, 13 (4), 18–28. DOI 10.1109/5254.708428.

[45] Jakub Dupák. *Memory Safety Analysis in Rust GCC*. 2024. `http://hdl.handle.net/10467/113390`. Accessed: 30-04-2024.

Appendix A
Thesis assignment



CTU CZECH TECHNICAL UNIVERSITY IN PRAGUE

MASTER‘S THESIS ASSIGNMENT

I. Personal and study details

Student's name:	**Sláma Lukáš**	Personal ID number:	**483749**
Faculty / Institute:	**Faculty of Electrical Engineering**		
Department / Institute:	**Department of Computer Science**		
Study program:	**Open Informatics**		
Specialisation:	**Software Engineering**		

II. Master’s thesis details

Master’s thesis title in English:

Behavioral data processing

Master’s thesis title in Czech:

Zpracování behaviorálních dat

Guidelines:

The aim of this work is to explore the usefulness of behavioural data that can be collected passively during the use of various electronic devices for the same purpose.
1) Familiarize yourself with behavioral data collection and digital phenotyping.
2) Monitor 10 volunteers for at least 2 weeks, using one of the mobile platforms for behavioral data collection.
3) Analyze and visualize the behavioral data.

Bibliography / sources:

[1] Mood state prediction from speech of varying acoustic quality for individuals with bipolar disorder, Gideon, 2016
[2] Recognition of Depression in Bipolar Disorder: Leveraging Cohor and Person-Specific Knowledge, Khorram, 2016
[3] Ecologically valid long-term mood monitoring of individuals with bipolar disorder using speech, Zahi N Karam 2014
[4] Realizing the potential of mobile mental health: New methods for new data in psychiatry, John Torous, 2015

Name and workplace of master’s thesis supervisor:

doc. Ing. Daniel Novák, Ph.D. Analysis and Interpretation of Biomedical Data FEE

Name and workplace of second master’s thesis supervisor or consultant:

Date of master’s thesis assignment: **13.02.2023** Deadline for master's thesis submission: **09.01.2024**

Assignment valid until: **22.09.2024**

doc. Ing. Daniel Novák, Ph.D.	Head of department’s signature	prof. Mgr. Petr Páta, Ph.D.
Supervisor’s signature		Dean’s signature

III. Assignment receipt

The student acknowledges that the master’s thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the master’s thesis, the author must state the names of consultants and include a list of references.

.

Date of assignment receipt Student’s signature

CVUT-CZ-ZDP-2015.1 © ČVUT v Praze, Design: ČVUT v Praze, VIC

Appendix B
Used software

In accordance with the *Methodological guideline No. 5/2023*[1] the following software was used in the development of this thesis:

- Microsoft Copilot (formerly Bing AI)[2] was used to explain certain unfamiliar topics or expressions. As for help with programming, Copilot was used to explain unfamiliar concepts in languages that I was not very familiar with, such as Matlab or Python.
- Perplexity AI[3] was used to find relevant sources (studies, articles) on a given topic.
- DeepL[4] was used for grammar and spelling checking and for text style feedback and rephrasing suggestions.

The credit for the idea, style and phrasing of this appendix goes to Jakub Dupák [45].

[1] `https://www.cvut.cz/sites/default/files/content/d1dc93cd-5894-4521-b799-c7e715d3c59e/en/20231003-methodological-guideline-no-52023.pdf`
[2] `https://www.bing.com/chat`
[3] `https://www.perplexity.ai/`
[4] `https://www.deepl.com/write`

Appendix C
Qualix source code

The source code for the Qualix application is freely available at `https://github.com/lukislama/Qualix` under a GPL-3.0 license.

Appendix D
Build, deployment and use guide

D.1 Introduction

Qualix build and deployment process is designed to be as simple as possible, while offering enough customization options to meet various preferences in terms of deployment targets and feature sets. Qualix is designed to run in Docker containers, which can be deployed locally or on any major cloud provider. Here I will only provide instructions on how to deploy Qualix locally, since there are several cloud providers and each has a unique way of running Docker containers. For a guide on how to setup Docker on popular cloud providers, see article[5] by pandaquests or official documentation for AWS[6], Azure [7] or Google Cloud[8].

D.2 Build and deployment

D.2.1 Introduction

Before building and deploying Qualix, you may want to change some basic settings, such as the login information or which port to expose.

Qualix offers two users by default, a basic user "user" and an administrative user "admin". The passwords are the same as the usernames. The difference is that the administrative user can access the Settings view and the List view containing information about participants in a given LAMP study. To change the login information from this basic setting you can modify the *SecurityConfig.java* file located at *src/main/java/com/example/application/security/SecurityConfig.java*

To change the exposed port, you can modify the Docker compose file located in the root directory. If you want Qualix to be accessible via the port 5500 for example, you would modify the `ports` section of the Docker compose file like so:

```
ports:
  - "5500:8080"
```

D.2.2 Requirements

Apart from Docker, this project requires the following packages:

- Maven
- JDK 17

[5] https://medium.com/codex/using-docker-on-popular-cloud-platforms-like-aws-google-cloud-and-azure-fc1bc377830f
[6] https://docs.aws.amazon.com/AmazonECS/latest/developerguide/create-container-image.html
[7] https://learn.microsoft.com/en-us/azure/container-instances/
[8] https://cloud.google.com/build/docs/build-push-docker-image

To install Docker you can use the official Docker installation guide[9]. You can install the required packages using your preferred package manager, such as apt, yum, brew, etc. For example, on Ubuntu, you would run the following commands:

```
sudo apt update
sudo apt install maven openjdk-17-jdk
```

D.2.3 Installation

Once you have the prerequisites ready, you can use the *build_and_run.sh* script that is included in the root directory. This script will perform the following tasks:

1. Build the application service from the custom Dockerfile that is included in the root directory. The Dockerfile specifies how to create the image, which will contain the Java application, the Python scripts, and the necessary dependencies.
2. Build the database service from the postgres:16.1 image, which is a pre-built image that contains the PostgreSQL database server.
3. Run the application and the database containers, exposing port 8080 to the host machine and linking them with the environment variables.
4. Launch the application on the container, which will be accessible from your browser at *http://localhost:8080*.

To run the script, open a terminal window and navigate to the root folder of this repository. Then, execute the following command:

```
sudo ./build_and_run.sh
```

D.2.4 Initial setup

D.2.4.1 Connecting to a LAMP server

In order for Qualix to communicate with a LAMP server, you need to enter the credentials for your LAMP server. You can do this before building and deploying Qualix by filling in the prepared fields in the *application.properties* file located at *src/main/resources/application.properties*, or after building and deploying in the Settings tab of the application. If you choose to modify the application properties file, be sure to also set *Qualix.server-set=true* like so:

```
Qualix.lamp-server-address=YOUR LAMP SERVER ADDRESS
Qualix.lamp-access-key=YOUR LAMP ACCESS KEY
Qualix.lamp-secret-key=YOUR LAMP SECRET KEY
Qualix.lamp-study-id=YOUR LAMP STUDY ID

Qualix.server-set=true
```

D.2.4.2 Setting up email notifications

Qualix can send you daily email notifications regarding the quality of gathered data from the previous day. In order to use this feature you need to use a Google account with 2FA enabled and generate a new *App password*. Instructions on how to create a Google app password can be found here[10]. Do NOT use your Google account password. Qualix will not access any information on your Google account, it will only use it to

[9] https://docs.docker.com/engine/install/
[10] https://support.google.com/accounts/answer/185833?hl=en&sjid=3303992743476717718-EU

send notification emails. Still, it is recommended to create a new Google account if you wish to use this feature.

Same as with the LAMP server credentials, you can enter your Google account information in the *application.properties* file or in the Settings tab. If you choose to modify the application properties file, be sure to also set *Qualix.email-set=true* like so:

```
Qualix.google-email-address=YOUR GOOGLE EMAIL ADDRESS
Qualix.google-app-password=YOUR GENERATED GOOGLE APP PASSWORD - DO NOT
USE YOUR GOOGLE ACCOUNT PASSWORD
Qualix.recipient-email-address=EMAIL ADDRESS THAT WILL RECEIVE
NOTIFICATIONS

Qualix.email-set=true
```

D.3 Using Qualix

Here you will find a step-by-step tutorial on how to use Qualix and how to set it up using the GUI part of the application.

After deploying Qualix and connecting to it, you will see a login screen — image D.1.

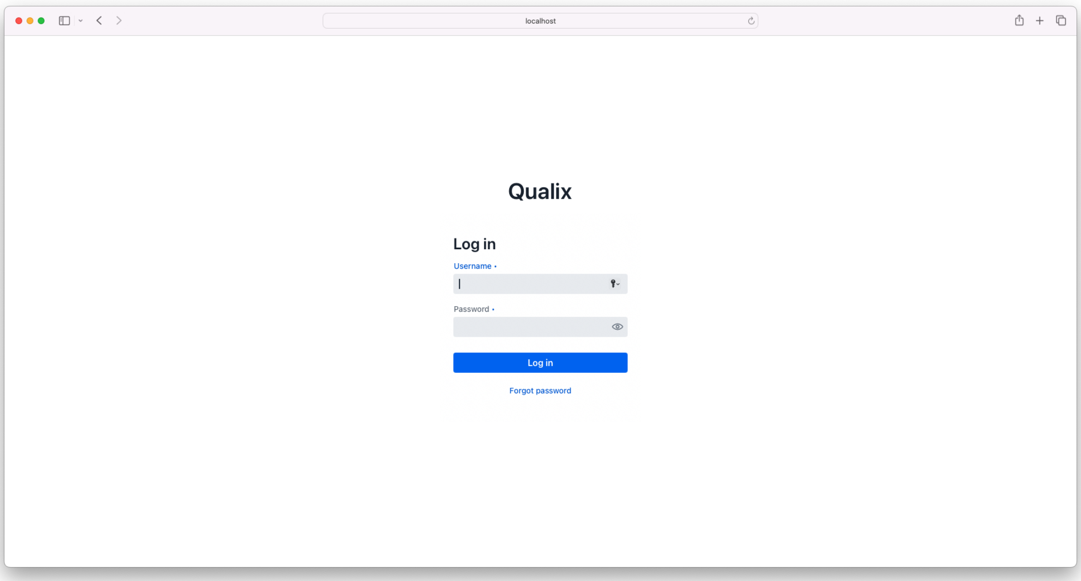


Figure D.1. Qualix login screen.

Login with an account that has administrative privileges, and you will be greeted with an empty Dashboard view — image D.2.

Dashboard view is the main view of the application, which will show you information about the quality of collected data at a glance.

To navigate Qualix, you can use the menu on the left side of the screen. To log out of Qualix, use the *Log-out button* on the upper right hand side of the screen.

Now navigate to the List view by clicking on *List* in the view menu — image D.3.

List view is only accessible to an account with administrative access. On this view, personal information about your participants can be stored. Now click on *Settings* —

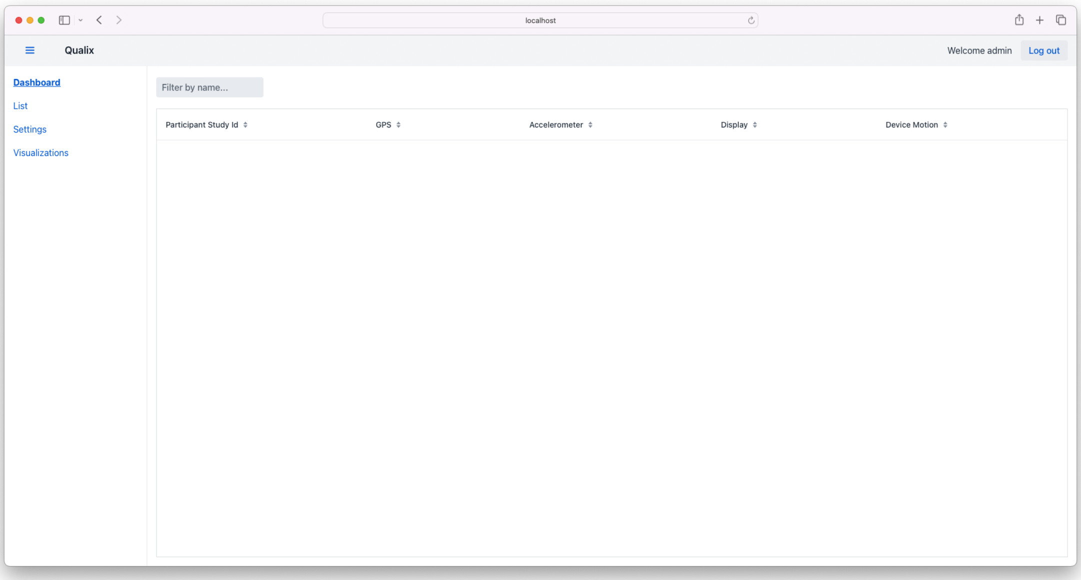


Figure D.2. Qualix empty Dashboard view.

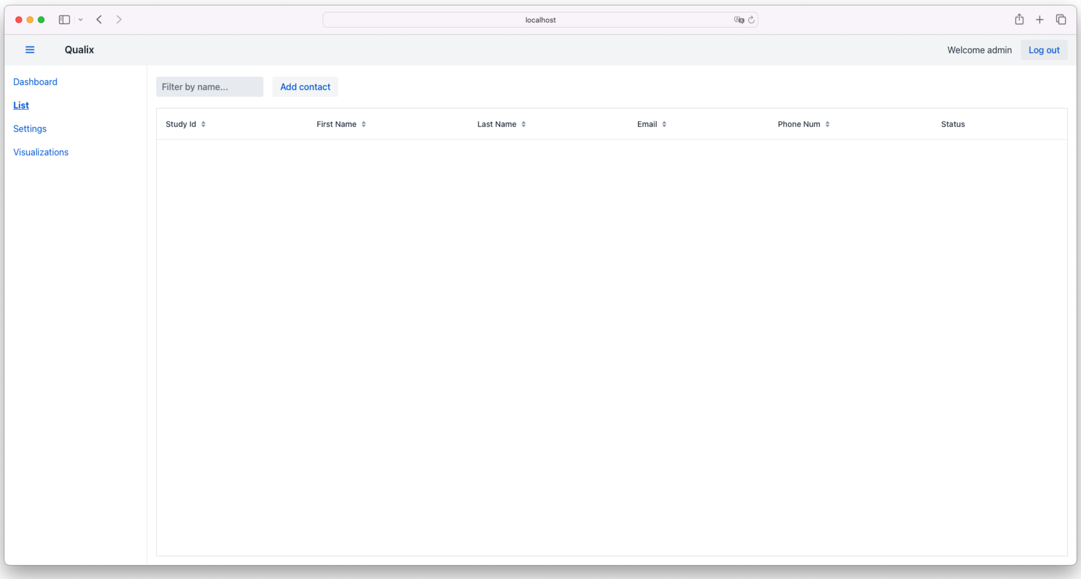


Figure D.3. Qualix empty List view.

image D.4, and we will set up connection to your LAMP server and Google account for sending notification emails.

Settings view can be used to set up connection to your LAMP server, Google account, and for creating a data cache. The data cache works by downloading data from all participants in a given study from your LAMP server over the last week. After creating the data cache, Qualix is set up to automatically download the latest data from the previous day, and to delete data older than 7 days, so that a data buffer of 7 days is always kept. This happens automatically every night. Note that this will not delete data from your LAMP server, only from the local data cache of Qualix. This data cache is then used for fast data processing and visualization inside Qualix.

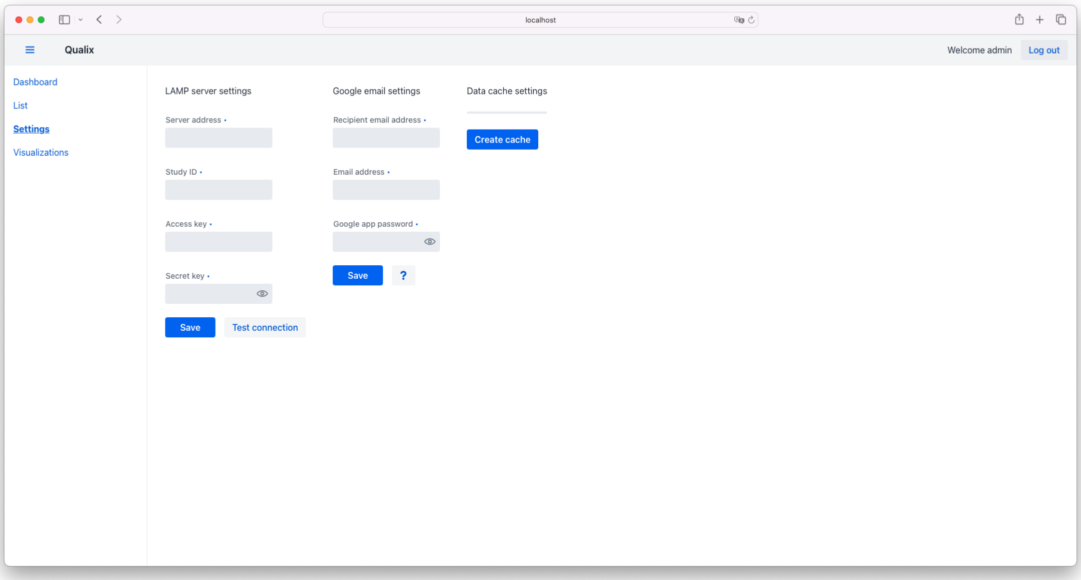
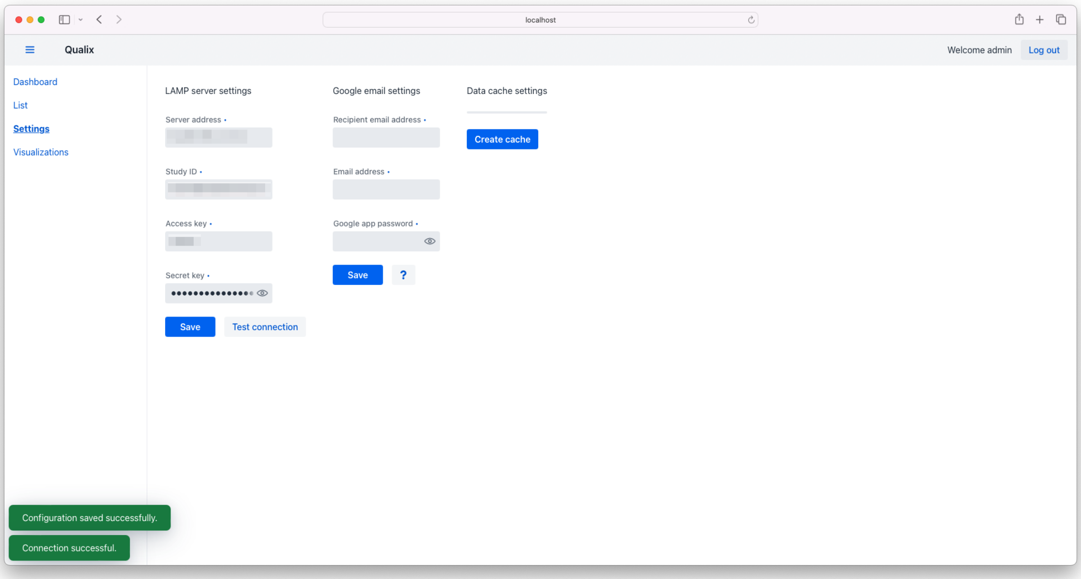


Figure D.4. Qualix empty Settings view.



Configuration saved successfully.

Connection successful.

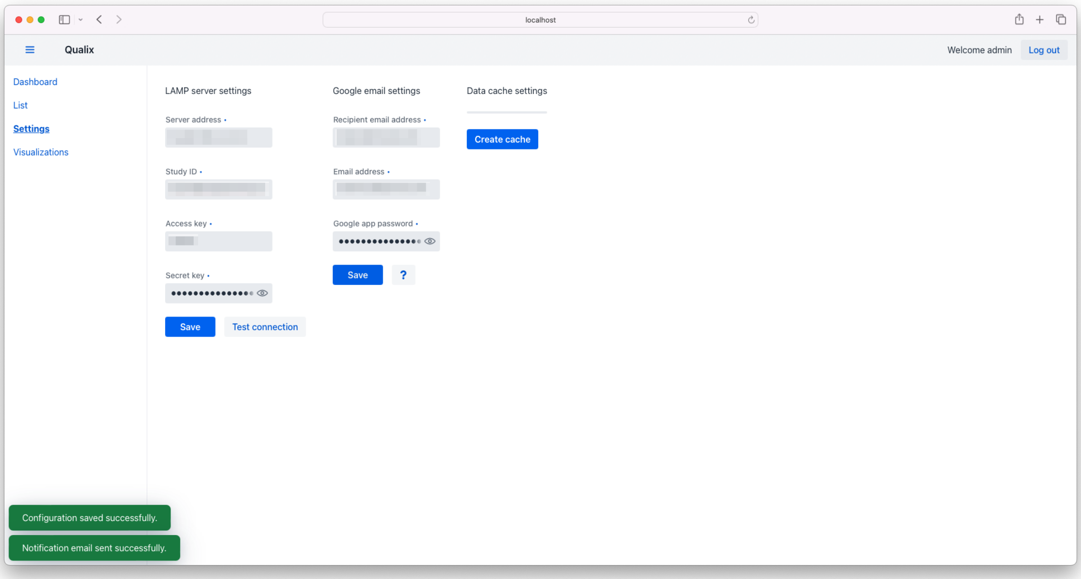
Figure D.5. Qualix Settings view - LAMP server set.

Now we will set the LAMP server settings. Fill in the required information and click on the *Save* button under *LAMP server settings*. Qualix will then try to connect to your LAMP server and if it succeeds, display the following message — image D.5.

If you wish to test the connection to your LAMP server in the future, you can use the *Test connection* button. Now we will set up your Google email settings. You can click on the *question mark* button to learn more about Google app passwords. After filling in your information, click on *Save* under *Google email settings*. Qualix will now try to send a notification email to the recipient address. If it succeeds, you will receive a notification email that looks like so:

```
Your email has been set to receive notifications from the Qualix
application.
If you think this is a mistake, please contact the study
administrators at YOUR GOOGLE EMAIL ADDRESS.
```

And Qualix will display the following message — image D.6.

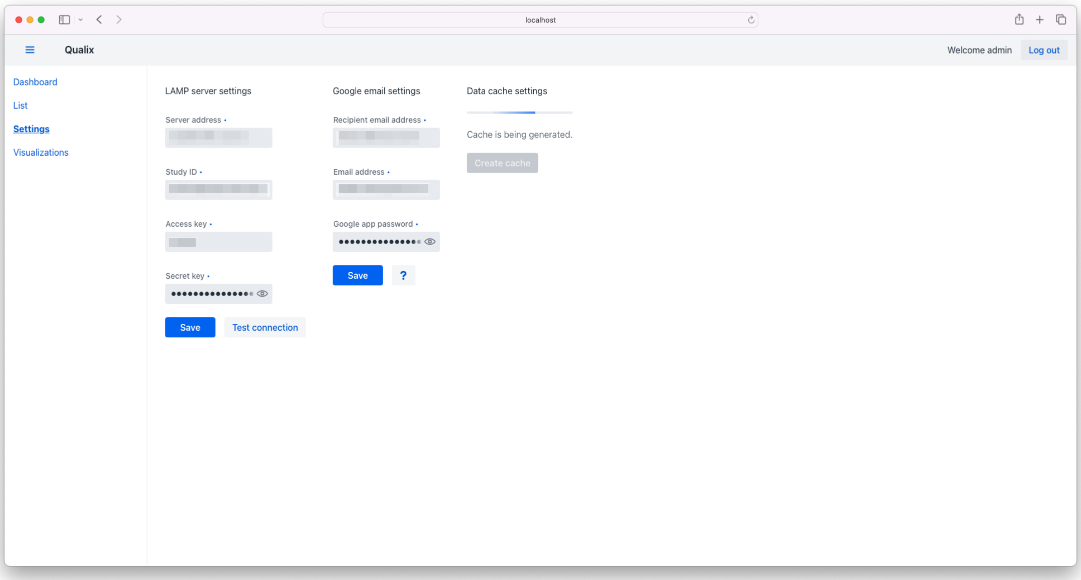


Configuration saved successfully.

Notification email sent successfully.

Figure D.6. Qualix Settings view - Email set.

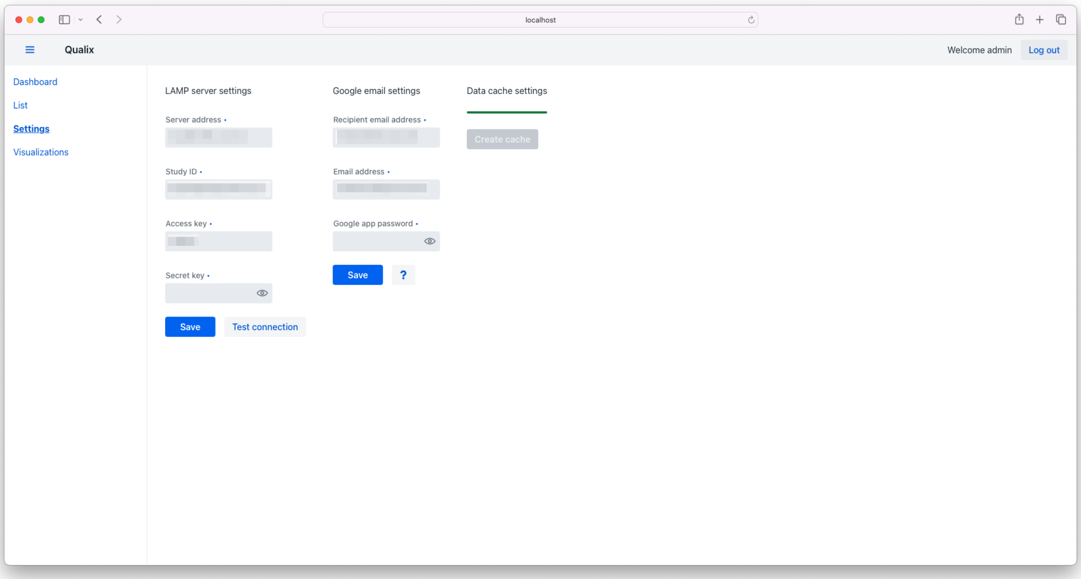
If you have set up your LAMP server settings and successfully connected to it, you can now create the data cache. This process can take up to multiple hours, if you have a high number of study participants (a few dozen). After clicking *Create cache*, the process will begin, and you will see a moving progress bar — image D.7.



Cache is being generated.

Figure D.7. Qualix Settings view - Building data cache.

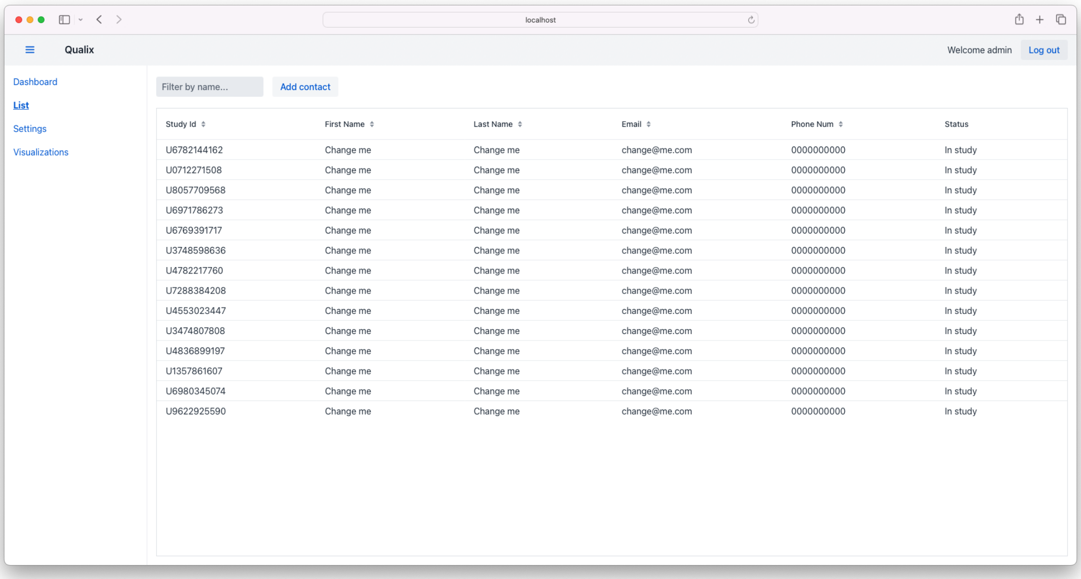
The data cache creation process runs on a background thread, so you can get out of the Settings view or even log out of Qualix and the data cache will still be created in the background. After the data cache is created, you will see a green line in the progress bar — image D.8.



Server address • [redacted]
Study ID • [redacted]
Access key • [redacted]
Secret key •
Recipient email address • [redacted]
Email address • [redacted]
Google app password •

Figure D.8. Qualix Settings view - Data cache built.

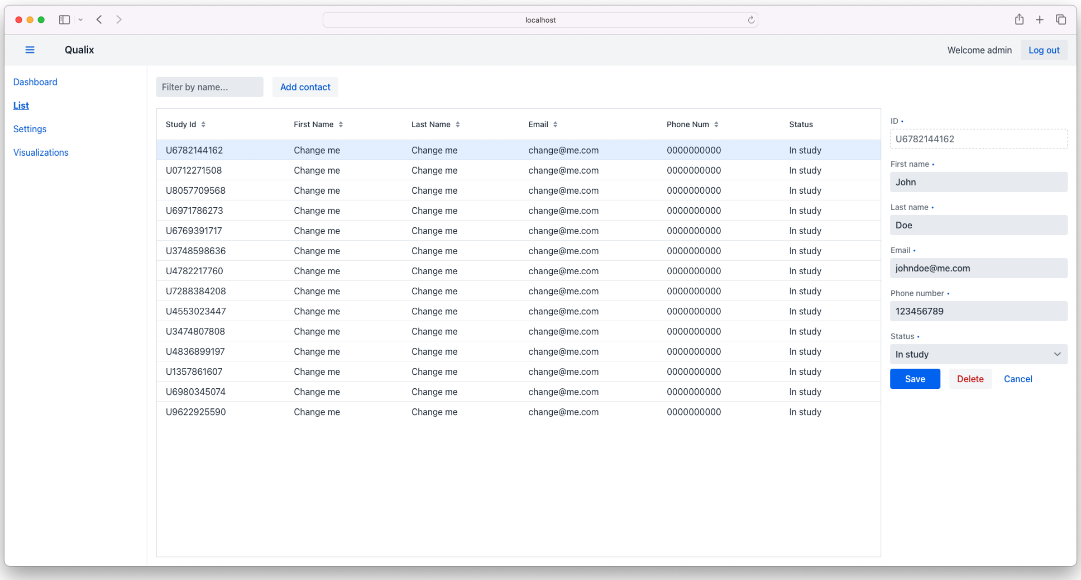
All functions of Qualix are now ready to be used. Navigate back to the List view and you will see a table like this — image D.9.



Study Id	First Name	Last Name	Email	Phone Num	Status
U6782144162	Change me	Change me	change@me.com	0000000000	In study
U0712271508	Change me	Change me	change@me.com	0000000000	In study
U8057709568	Change me	Change me	change@me.com	0000000000	In study
U6971786273	Change me	Change me	change@me.com	0000000000	In study
U6769391717	Change me	Change me	change@me.com	0000000000	In study
U3748598636	Change me	Change me	change@me.com	0000000000	In study
U4782217760	Change me	Change me	change@me.com	0000000000	In study
U7288384208	Change me	Change me	change@me.com	0000000000	In study
U4553023447	Change me	Change me	change@me.com	0000000000	In study
U3474807808	Change me	Change me	change@me.com	0000000000	In study
U4836899197	Change me	Change me	change@me.com	0000000000	In study
U1357861607	Change me	Change me	change@me.com	0000000000	In study
U6980345074	Change me	Change me	change@me.com	0000000000	In study
U9622925590	Change me	Change me	change@me.com	0000000000	In study

Figure D.9. Qualix List view - Before info change.

As you can see, all participants of your study have been loaded in. Since LAMP doesn't store personal information about its study participants, placeholder information has been generated for each study participant. If you wish to store personal information of your participants in Qualix, you can do so by changing the placeholder information. Click on a participant that you want to modify and a new menu will pop up — image D.10.



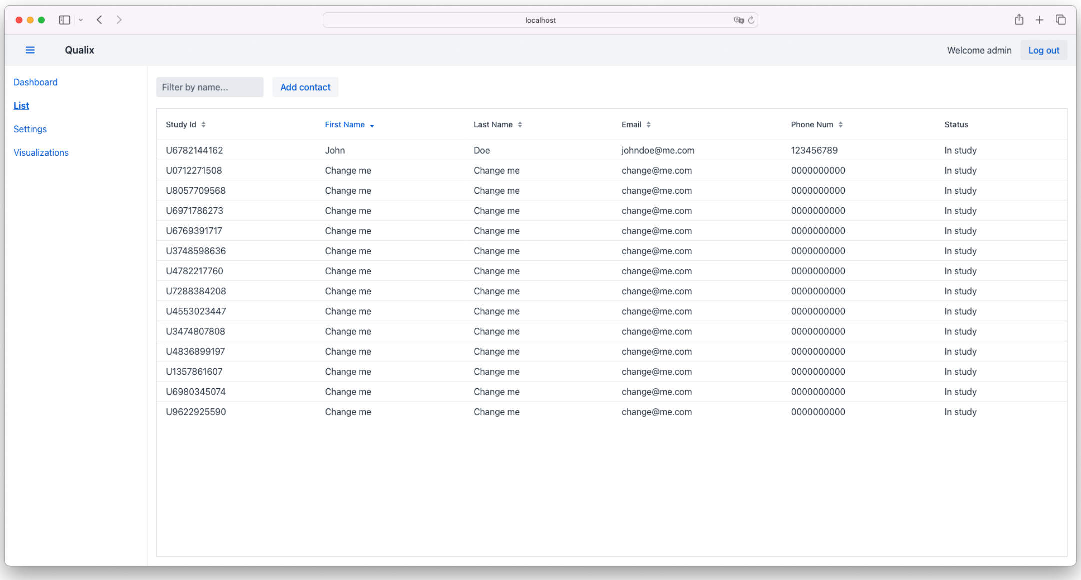
Study Id	First Name	Last Name	Email	Phone Num	Status
U6782144162	Change me	Change me	change@me.com	0000000000	In study
U0712271508	Change me	Change me	change@me.com	0000000000	In study
U8057709568	Change me	Change me	change@me.com	0000000000	In study
U6971786273	Change me	Change me	change@me.com	0000000000	In study
U6769391717	Change me	Change me	change@me.com	0000000000	In study
U3748598636	Change me	Change me	change@me.com	0000000000	In study
U4782217760	Change me	Change me	change@me.com	0000000000	In study
U7288384208	Change me	Change me	change@me.com	0000000000	In study
U4553023447	Change me	Change me	change@me.com	0000000000	In study
U3474807808	Change me	Change me	change@me.com	0000000000	In study
U4836899197	Change me	Change me	change@me.com	0000000000	In study
U1357861607	Change me	Change me	change@me.com	0000000000	In study
U6980345074	Change me	Change me	change@me.com	0000000000	In study
U9622925590	Change me	Change me	change@me.com	0000000000	In study

ID • U6782144162
First name • John
Last name • Doe
Email • johndoe@me.com
Phone number • 123456789
Status • In study

Save Delete Cancel

Figure D.10. Qualix List view - Changing info.

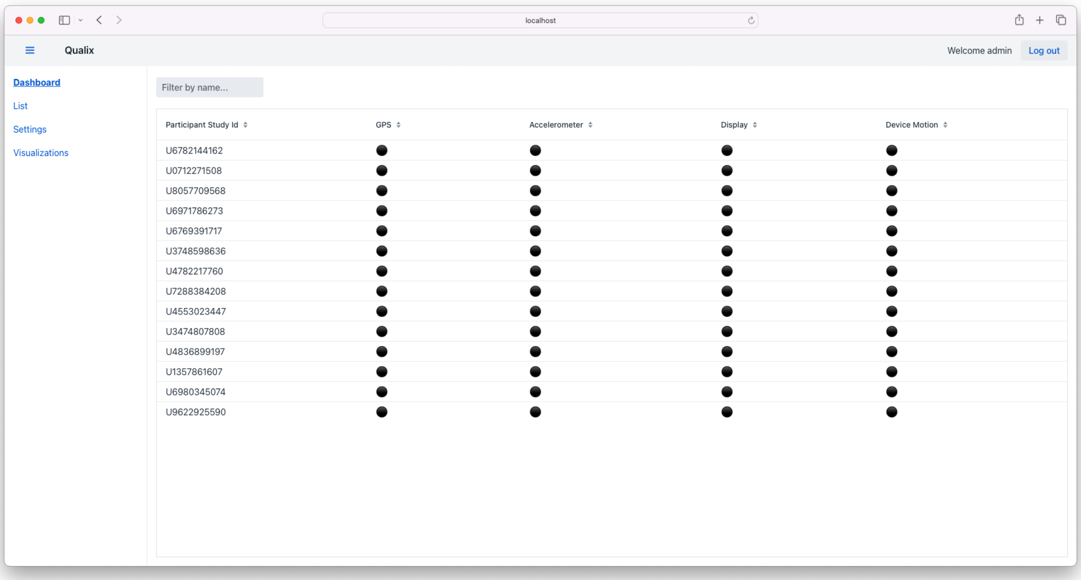
Here you can change the name, email, phone number and study status of a given participant, or delete the participant's information altogether. After making your changes, click on *Save*. After that, the new information will be saved — image D.11.



Study Id	First Name	Last Name	Email	Phone Num	Status
U6782144162	John	Doe	johndoe@me.com	123456789	In study
U0712271508	Change me	Change me	change@me.com	0000000000	In study
U8057709568	Change me	Change me	change@me.com	0000000000	In study
U6971786273	Change me	Change me	change@me.com	0000000000	In study
U6769391717	Change me	Change me	change@me.com	0000000000	In study
U3748598636	Change me	Change me	change@me.com	0000000000	In study
U4782217760	Change me	Change me	change@me.com	0000000000	In study
U7288384208	Change me	Change me	change@me.com	0000000000	In study
U4553023447	Change me	Change me	change@me.com	0000000000	In study
U3474807808	Change me	Change me	change@me.com	0000000000	In study
U4836899197	Change me	Change me	change@me.com	0000000000	In study
U1357861607	Change me	Change me	change@me.com	0000000000	In study
U6980345074	Change me	Change me	change@me.com	0000000000	In study
U9622925590	Change me	Change me	change@me.com	0000000000	In study

Figure D.11. Qualix List view - After info change.

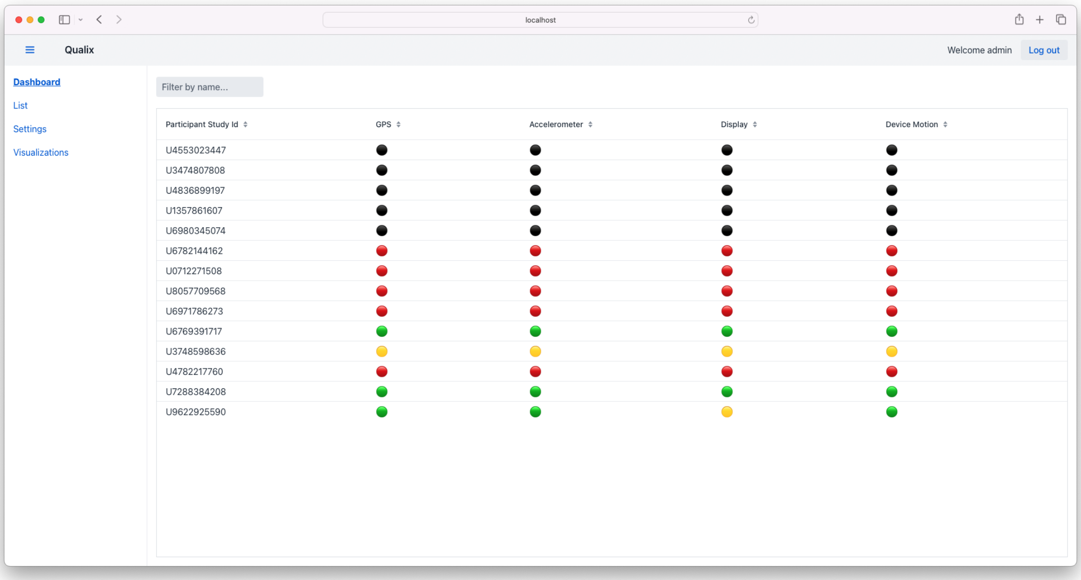
Now navigate to the main Dashboard view — image D.11.



Participant Study Id	GPS	Accelerometer	Display	Device Motion
U6782144162	●	●	●	●
U0712271508	●	●	●	●
U8057709568	●	●	●	●
U6971786273	●	●	●	●
U6769391717	●	●	●	●
U3748598636	●	●	●	●
U4782217760	●	●	●	●
U7288384208	●	●	●	●
U4553023447	●	●	●	●
U3474807808	●	●	●	●
U4836899197	●	●	●	●
U1357861607	●	●	●	●
U6980345074	●	●	●	●
U9622925590	●	●	●	●

Figure D.12. Qualix Dashboard view - Before data check.

This is what the Dashboard view will look like after first setting up the application. Qualix has downloaded the participants' ids from your LAMP server and is now checking the data quality from the previous day. After the check is finished, the Dashboard will look something like this — image D.13.



Participant Study Id	GPS	Accelerometer	Display	Device Motion
U4553023447	●	●	●	●
U3474807808	●	●	●	●
U4836899197	●	●	●	●
U1357861607	●	●	●	●
U6980345074	●	●	●	●
U6782144162	🔴	🔴	🔴	🔴
U0712271508	🔴	🔴	🔴	🔴
U8057709568	🔴	🔴	🔴	🔴
U6971786273	🔴	🔴	🔴	🔴
U6769391717	🟢	🟢	🟢	🟢
U3748598636	🟡	🟡	🟡	🟡
U4782217760	🔴	🔴	🔴	🔴
U7288384208	🟢	🟢	🟢	🟢
U9622925590	🟢	🟢	🟡	🟢

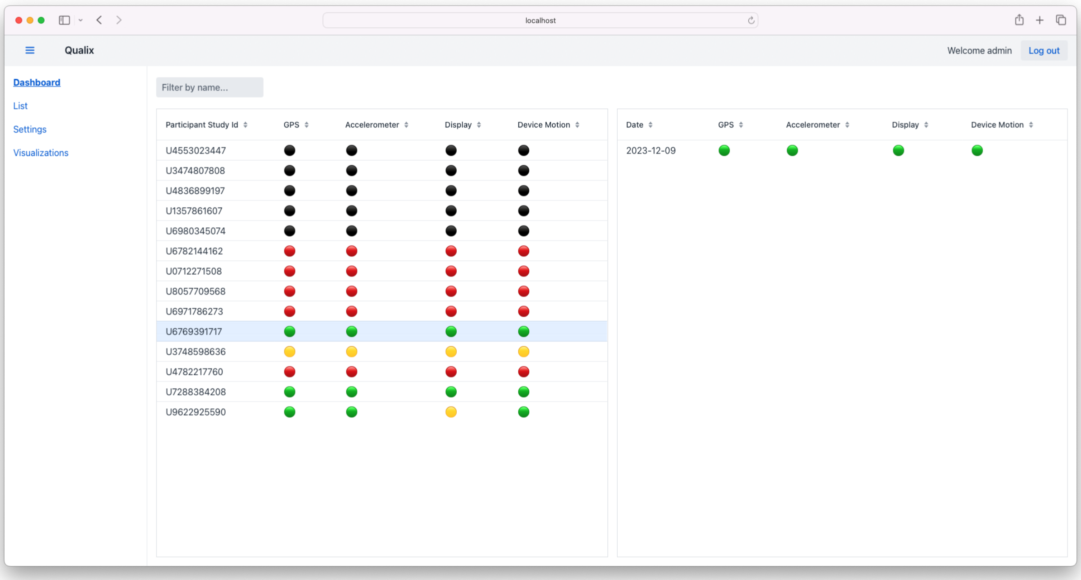
Figure D.13. Qualix Dashboard view - After data check.

Qualix is set up to give you information about the participants' data quality at a glance using the *semaphore* system:

- Green — Latest data arrived in the last 2 hours
- Yellow — Latest data arrived in the last 12 hours
- Red — Latest data arrived more than 12 hours ago

- Black — no data at all

Try clicking on some participant. A new table will pop up — image D.14.



Participant Study Id	GPS	Accelerometer	Display	Device Motion
U4553023447	⚫	⚫	⚫	⚫
U3474807808	⚫	⚫	⚫	⚫
U4836899197	⚫	⚫	⚫	⚫
U1357861607	⚫	⚫	⚫	⚫
U6980345074	⚫	⚫	⚫	⚫
U6782144162	🔴	🔴	🔴	🔴
U0712271508	🔴	🔴	🔴	🔴
U8057709568	🔴	🔴	🔴	🔴
U6971786273	🔴	🔴	🔴	🔴
U6769391717	🟢	🟢	🟢	🟢
U3748598636	🟡	🟡	🟡	🟡
U4782217760	🔴	🔴	🔴	🔴
U7288384208	🟢	🟢	🟢	🟢
U9622925590	🟢	🟢	🟡	🟢

Date	GPS	Accelerometer	Display	Device Motion
2023-12-09	🟢	🟢	🟢	🟢

Figure D.14. Qualix Dashboard view - More info.

This table will show you the history of the data quality for a selected participant. Now let's focus on the last view — Visualizations — image D.15.

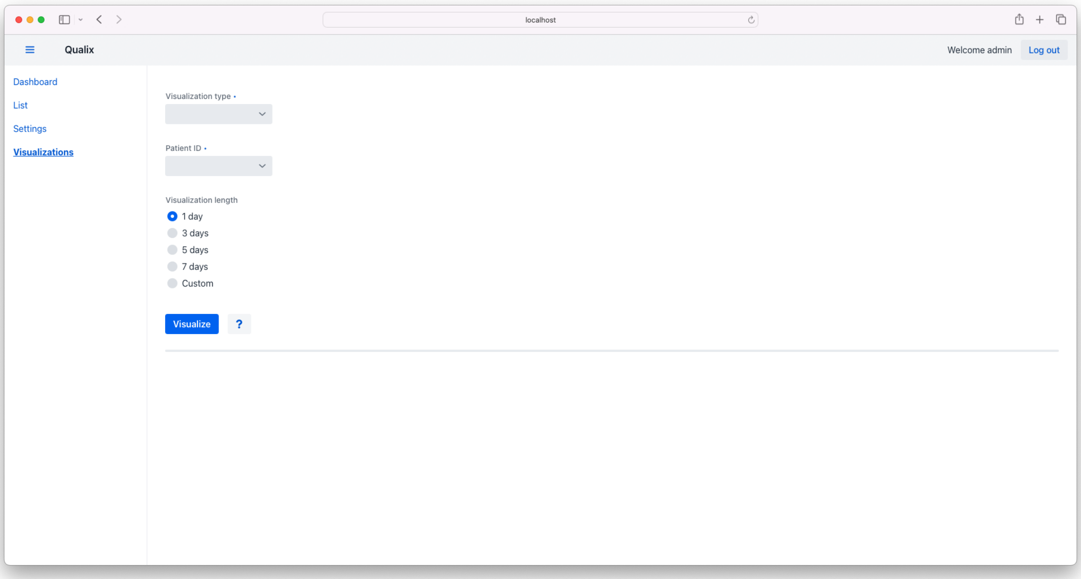


Figure D.15. Qualix - Empty Visualizations view.

This view utilizes the data cache function of Qualix to instantly visualize data for a selected participant. If you pick one of the provided visualization lengths, Qualix will use its cache function to generate a visualization instantly. If you choose the *Custom* setting, Qualix will download the necessary data from the LAMP server, then process them and then visualize them. This can take some time. It is therefore recommended to choose one of the provided visualization lengths.

Let's visualize data quality from the last day for some participant. Select the *Data_quality* option from the *Visualization type* menu, a patient you wish to visualize, select *1 day* as the visualization length and click on *Visualize* — image D.16.

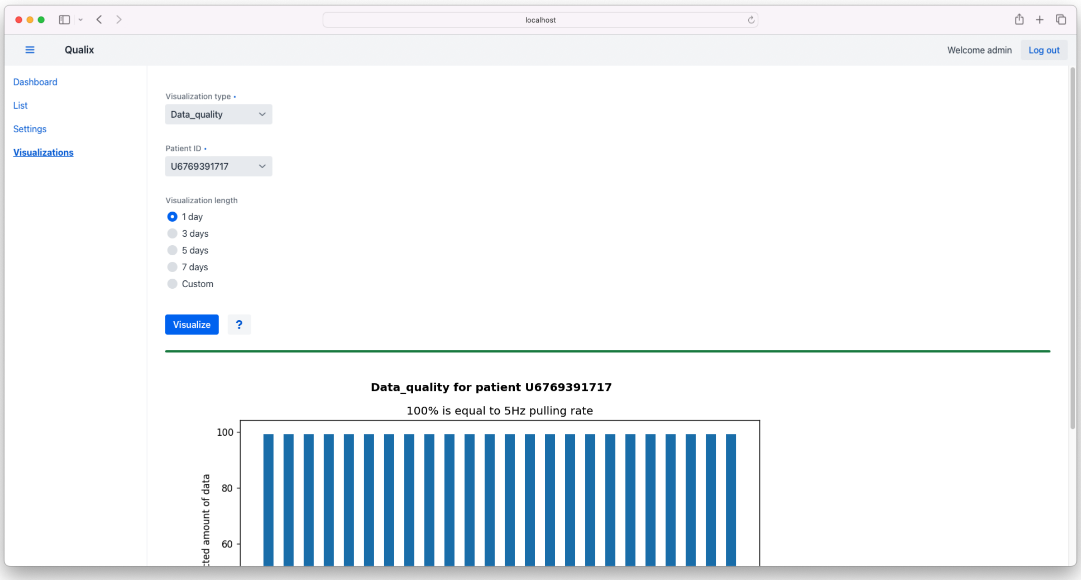
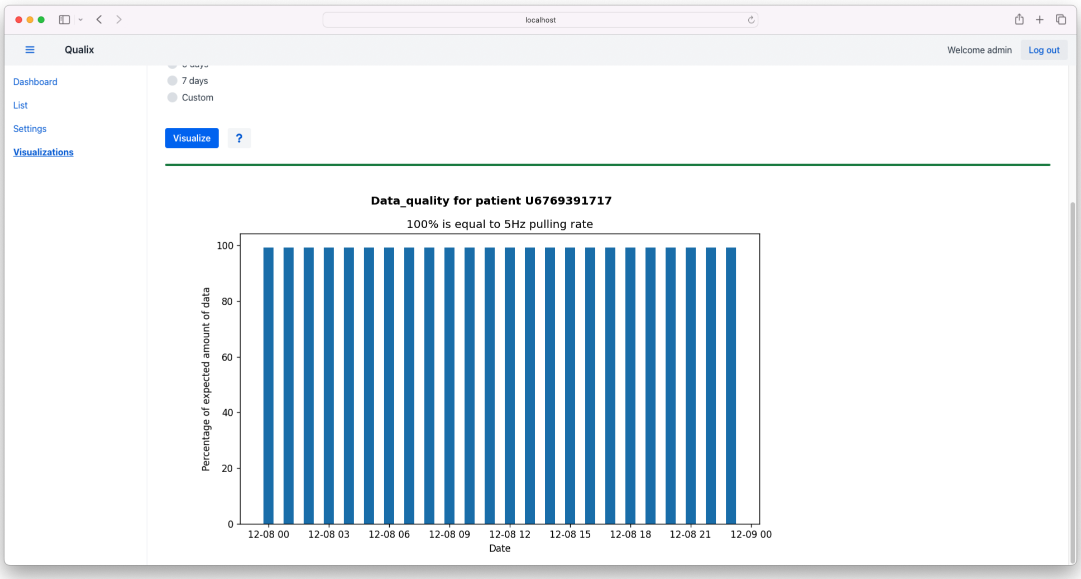


Figure D.16. Qualix Visualizations view - Data quality visualization.

You can now scroll the page down to see the full graph — image D.17.



localhost
Qualix
Welcome admin
Log out
Dashboard
List
Settings
Visualizations
7 days
Custom
Visualize
?
Data_quality for patient U6769391717
100% is equal to 5Hz pulling rate
Percentage of expected amount of data
Date

Figure D.17. Qualix Visualizations view - Full data quality visualization.

Since the generated graph is inserted as a simple image, you can download it by right-clicking on it and saving the image. Let's check another type of visualization.

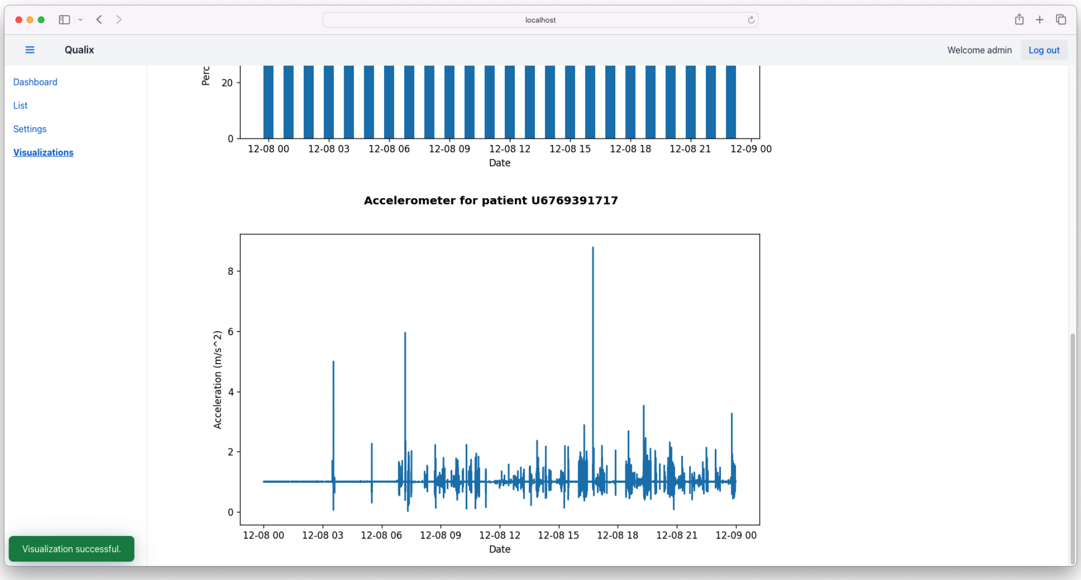


Figure D.18. Qualix Visualizations view - Full accelerometer visualization.

Select *Accelerometer* from the *Visualization type* menu and click on *Visualize* again. A new graph will be generated below the original one — image D.18.

Now let's check the *Custom* visualization length setting. After selecting it a new date time pickers will appear. Here you will be able to select a custom start and end of the visualization — images D.19 and D.20.

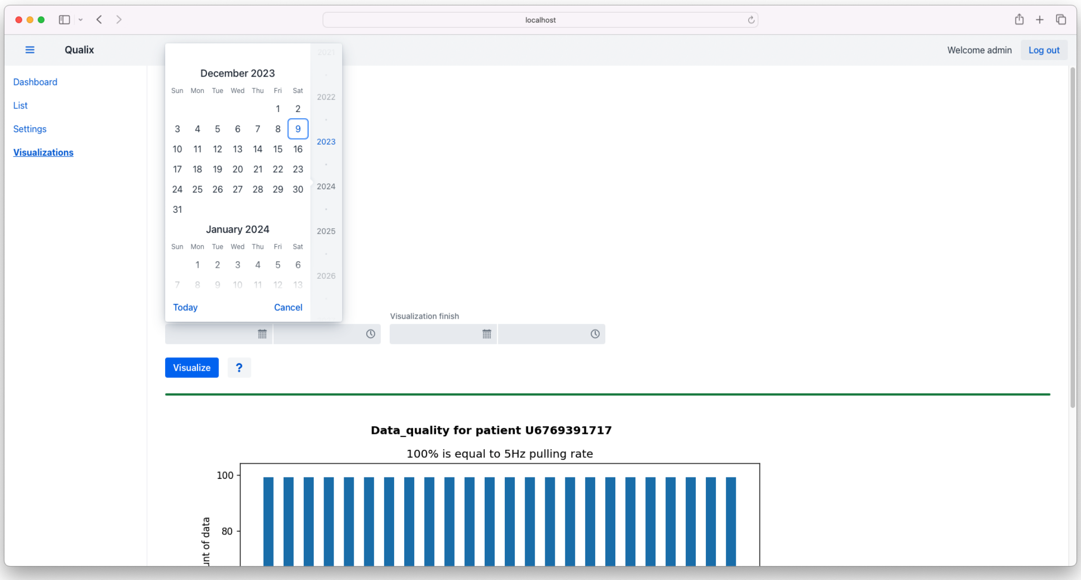
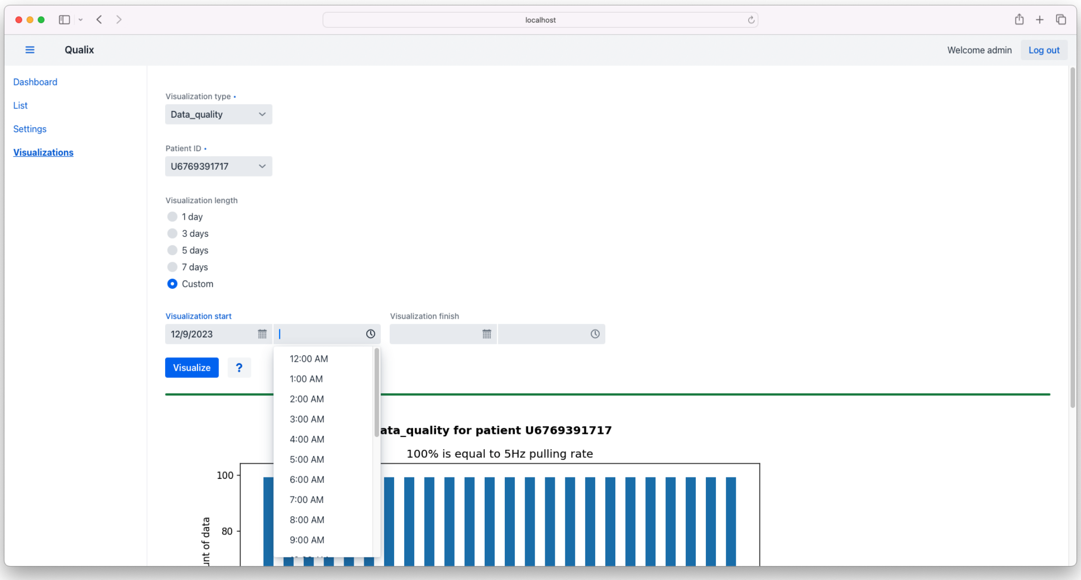


Figure D.19. Qualix Visualizations view - Custom visualization settings - date.



Visualization type • Data_quality

Patient ID • U6769391717

Visualization length
1 day
3 days
5 days
7 days
[✓ Custom]

Visualization start 12/9/2023

Visualization finish

ata_quality for patient U6769391717

100% is equal to 5Hz pulling rate

unt of data

Figure D.20. Qualix Visualizations view - Custom visualization settings - time.

Appendix E
Glossary

CBT	■	Cognitive Behavioral Therapy
GUI	■	Graphical User Interface
HBHU	■	Healthy Body Healthy U
IA	■	Interned Addiction
LAMP	■	Learn, Assess, Manage and Prevent
LDAP	■	Lightweight Directory Access Protocol
LMIC	■	Low -and Middle-Income Countries
L5	■	Least active Five-hour period or nocturnal activity
MVC	■	Model, View, Controller
M10	■	Most active Ten-hour period or daytime activity
RBF	■	Radial Basis Function
RCT	■	Randomized Controlled Trial
RMS	■	Root Mean Square
rTMS	■	repetitive Trans-cranial Magnetic Stimulation
SVM	■	Support Vector Machine