University of Zurich[UZH]

ASSIGNMENT – Communication Systems Group, Prof. Dr. Burkhard Stiller

WebMaDa Extension Addressing Transparency Request for Data Owners

Neva Silvestri
Waltenschwil, Schweiz
Student ID: 12-921-219

Supervisor: Corinna Schmitt, Sina Rafati
Date of Submission: July 4, 2017

University of Zurich
Department of Informatics (IFI)
Binzmühlestrasse 14, CH-8050 Zürich, Switzerland

CSG
Department of Informatics
Communication Systems Group

Assignment
Communication Systems Group (CSG)
Department of Informatics (IFI)
University of Zurich
Binzmühlestrasse 14, CH-8050 Zürich, Switzerland
URL: http://www.csg.uzh.ch/

Zusammenfassung

In diesem Report geht es um die Entwicklung eines benutzerfreundlichen Userinterfases für eine transparente Anfrage eines WSN Owners in WebMaDa. Der WSN Owner hat bereits ein anderes Interface für seine WSNs. Dieses Interface wird nun durch einen weiteren Knopf erweitert, nämlich Filtering Options". Dadurch wird es dem WSN Owner möglich sein zu sehen, welche Daten von den autorisierten Benutzern seines WSNs gepusht oder gepullt werden. Die Optionen beim Filtern werden vorallem den User, das Datum, ob Push oder Pull und die Daten selbst beinhalten. Die Entwicklung wird in HTML mit Smarty fÃ¼r das Frontend und mit PHP im Backend, welches vorallem die Anfragen zur MySQL Datenbank beinhaltet, stattfinden. Wenn nun ein WSN Owner die gewünschten Filteroptionen gesetzt hat, kann er danach das Ergebnis in einem PDF abspeichern. Die Evaluation der Arbeit wird durch einen Proof of Operability stattfinden.

Abstract

This report is about the development of a user-friendly user interface for the transparency request of a WSN owner in WebMaDa. The WSN owner already has different options for his/her WSNs. This will be extended by another option: Filtering Options. Through this interface a WSN Owner will be able to see what authorized users in his WSN pushed or pulled. The filtering options will contain fields like user, date, if push or pull and data. The development will be in HTML with Smarty for the front-end and PHP for the back-end, which means for the queries to the MySQL database. When a WSN owner chose the filtering options that fit his/her interest he/she can afterwards save the results to a PDF. The evaluation will be through a Proof of Operability.

Acknowledgments

At this point I would like to thank my supervisor Dr. Corinna Schmitt for her positive being, her strong support and the time she was always willing to offer during this assignment. Also I would like to thank Prof. Dr. Burkhard Stiller for the possibility to complete this assignment at the Communications Systems Group in the Department of Computer Science of the University of Zurich.

Contents

Chapter 1

Introduction

1.1 Motivation

Due to the growth of the Internet and the device diversity together with their communication capability the Internet of Things (IoT) is a hot topic. The Internet of Things is not limited to Peer-to-Peer (P2P) networks and devices like server, computers, and routers any more. It also includes wireless sensor devices connected in a Wireless Sensor Network (WSN) [1]. The application range goes from intelligent homes, logistics, health care to environmental monitoring. All applications have in common a huge amount of collector sensor data (e.g., temperature, brightness, humidity) under different operating systems. The collected data also includes sensitive data and might be accessed by authorized persons. Data owner have concerns on following the data flow, especially who accessed which data and when. In order to address this transparency concern a solution must be developed giving the data owners the opportunity to follow this data access. The assignment this assignment will be built on is called SecureWSN [3]. It allows data collection and visualization of data for constraint devices,. In the background a database is running logging collected data and supporting the access management. In the Front-end data owners have the opportunity to grant access rights and see collected data. The already established solution has the drawback that collected data is stored in a database in the back-end that can only be accessed by the database administrator who is a different person compared to the data owner. Further, the data owner is only able to see his collected data and granted rights to authorized users, but not when and which data was accessed. Currently two graphical user-interfaces are in place where the data owner can see his data. Those are CoMaDa [3] [5] and WebMaDa [4] that will undergo an extension to address the aforementioned transparency request of data owners.

1.2 Description of Work

The task of this assignment is to develop a user-friendly solution addressing the transparency request under WebMaDa. To reach this goal the following work needs to be done:

First, it needs to be analyzed what information is stored in the existing database using MySQL and how stored data is linked together, because for each network ten tables are stored with interrelated information as described in [4]. Second, it needs to be analyzed how to access this database in a secure manner and how data can be extracted and ported to WebMaDa to display it to the data owner. This will include decision taking concerning re-use/new creation of data transport solution and credential checks for the data owner. Next, the access and data transfer solution will be designed and implemented combined with a user-friendly GUI integrated in the existing WebMaDa. Important is that envisioned functionality is only available to network owners and not to authorized external viewers. This GUI needs to offer the data owner filtering options to request specified data (e.g., What data did user X access? Who accessed data of sensor X?). The resulting presentation of the data access should also indicate time-stamps of the access. Finally, the developed solution will be evaluated concerning work flow, filtering, and performance. The result of this assignment will be a user-friendly GUI extension for WebMaDa addressing the transparency request of data owners offering different filtering options allowing specific information requests.

1.3 Assignment Outline

The rest of this assignment paper is structured as follows. Chapter 2 is about related work where topics which are relevant to this work are looked deeper into. Chapter 3 presents the design decisions which are based on the beforehand chapter and contains details about the user-friendly GUI. Chapter 4 will show the implementation process. In chapter 5 details about the evaluation of the implementation will be given including a Proof of Operability. The final chapter, chapter 6, is the conclusion of this work.

Chapter 2

Related Work

This chapter is about topics that are relevant for this work. It will make the reader understand better where the idea of this work comes from, what is around it and how it all works. The first focus lies on the Secure Wireless Sensor Network (SecureWSN), which is a framework that is researched at the Communication Systems Group (CSG) at the University of Zurich (UZH) [6]. The next main topics will be the main parts of the SecureWSN, which are the Configuration, Management, and Data Handling Framework (CoMaDa) and the Mobile Access and Data Handling Framework (WebMaDa). Further also the existing WebMaDa database will be looked into.

2.1 SecureWSN

The SecureWSN started with the development of an efficient data transmission protocol within a Wireless Sensor Network (WSN). When this was concluded with TinyIPFIX the next step was to make the request of a user secure. Always having in mind to re-use standards from IP networks three security options were implemented that have authentication, session key agreement and being standard-based in common. They are called TinySAM protocol, TinyTO protocol and TinyDTLS protocol. After this goal was reached there had to be a GUI because in this time users like to configure things in handsome ways. The GUI that was designed is called Configuration, Management, and Data Handling Framework (CoMaDa). As the name suggests a user was able to configure, manage and handle data through CoMaDa. CoMaDa has one drawback though because the user is required to sit in front of the computer where CoMaDa is running. As this framework is all about wireless sensors, this seemed like a contradiction and so a web-based mobile access was designed and implemented which is called Mobile Access and Data Handling Framework (WebMaDa). Through this framework authorized users can access the data online. [2]
In Figure 2.1 and overview of the SecureWSN is given. WebMaDa can be found in the top right corner with its server setup, below a closer view of CoMaDa, how the functionalities are linked and how the data can be visualized and on the left side an data aggregation of a node is shown.

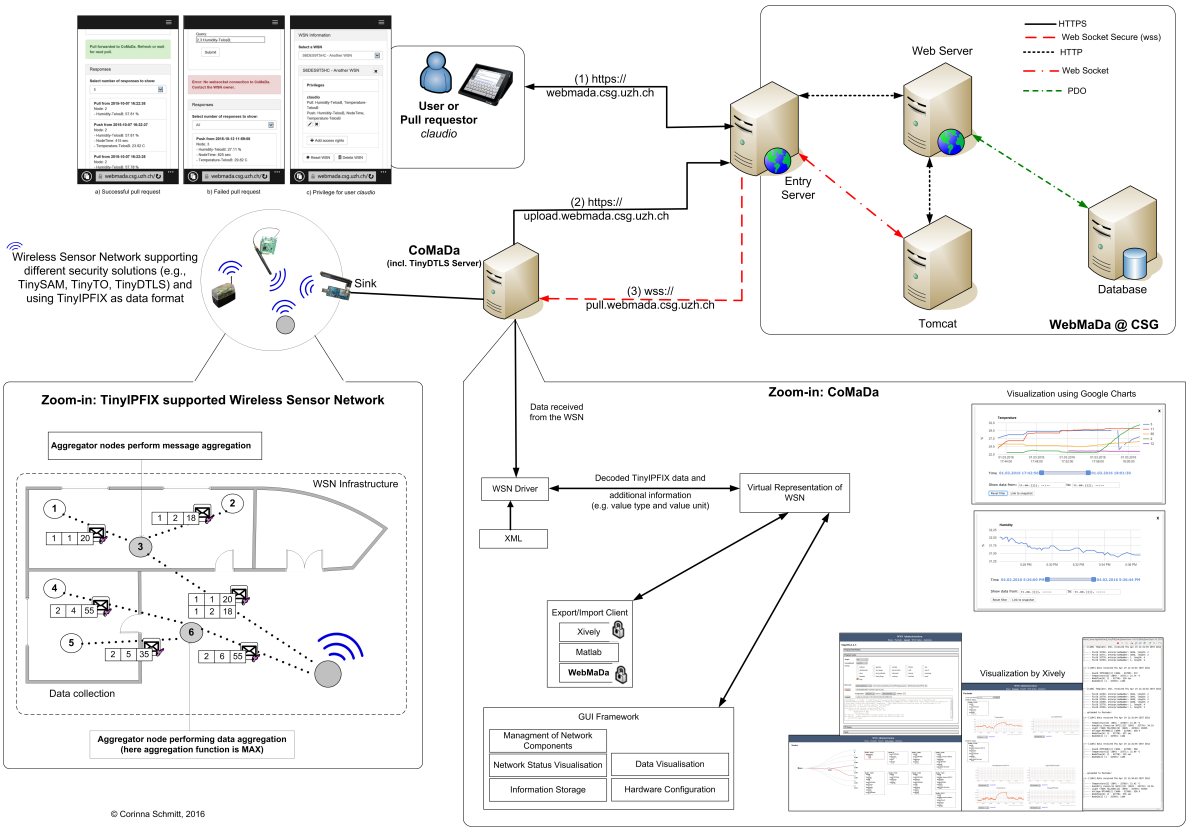

Figure 2.1: Overview SecureWSN [2]

2.1.1 CoMaDa

In CoMaDa a user can configure his/her WSN. In detail this means that nodes can be added to a WSN, they can be programmed and updated. Also the network status is always visible. The user has also the possibility to receive the data in a raw format or let it visualize in a curve design [2]. The code of CoMaDa is written mostly in Java. The project itself is called WSNDataFramework in Eclipse. CoMaDa has iteself a PostGreSQL database which is more or less a copy of the WebMaDa database. The difference is that in the CoMaDa database there are only the tables that are created for every WSN.

2.1.2 WebMaDa

As mentioned before CoMaDa had a drawback that it was not mobile. WebMaDa is the missing extension to this drawback. It is accessible through webmada.csg.uzh.ch. In WebMaDa it is possible to register a WSN that was created with the help of CoMaDa. You can give other registered users push and pull rights on your own WSN. Other users can also grant you rights to their WSNs. If you have the rights to pull data in a WSN you can go so in WebMaDa. There you can select e specific node and what data you want to pull, f.e. Humidity. What is selected can be added to a query which can be submitted if

satisfying. If the network is up and running the data that you we're looking for will be shown. It is also possible, like in CoMaDa, to visualize the data.

2.2 MySQL Database

WebMaDa has a MySQL database. There are five tables that save common data. Those tables are called ActiveWSN, which shows the WSNs that exist and their owner with name, username and password; Admin; InvitationCode; Rights, which shows the username that has rights on a certain WSN; User, which has personal information about the users with fields like username, password, firstname, lastname, email and affiliation.
For every WSN that is registered through WebMaDa 10 tables are created. DATA, TOPOLOGY, DataStream, Sensor, SensorToNodetype, DataRecord, Node, Report and Response. In Figure 2.2 an overview of the tables is given. The ten tables that are created for every WSN are in the box of "For every WSN these tables are created". There are three tables in grey, which are decoupled from the rest which means that they have no connection to another table via Primary Key, Foreign Key or shared keywords. In the second box, which is called "Common tables", the tables that store common data are seen in detail.
Overall keywords that are written in bold indicate a Primary Key. Keywords, which are written in italic indicate a Foreign Key. There are also keywords that are written boldly and in italic, which indicates a keyword that is both Primary and Foreign Key. Lines with an arrow at the end show the relationship from Primary Key to Foreign Key. Dashed lines however just show the relationship from the same keywords.

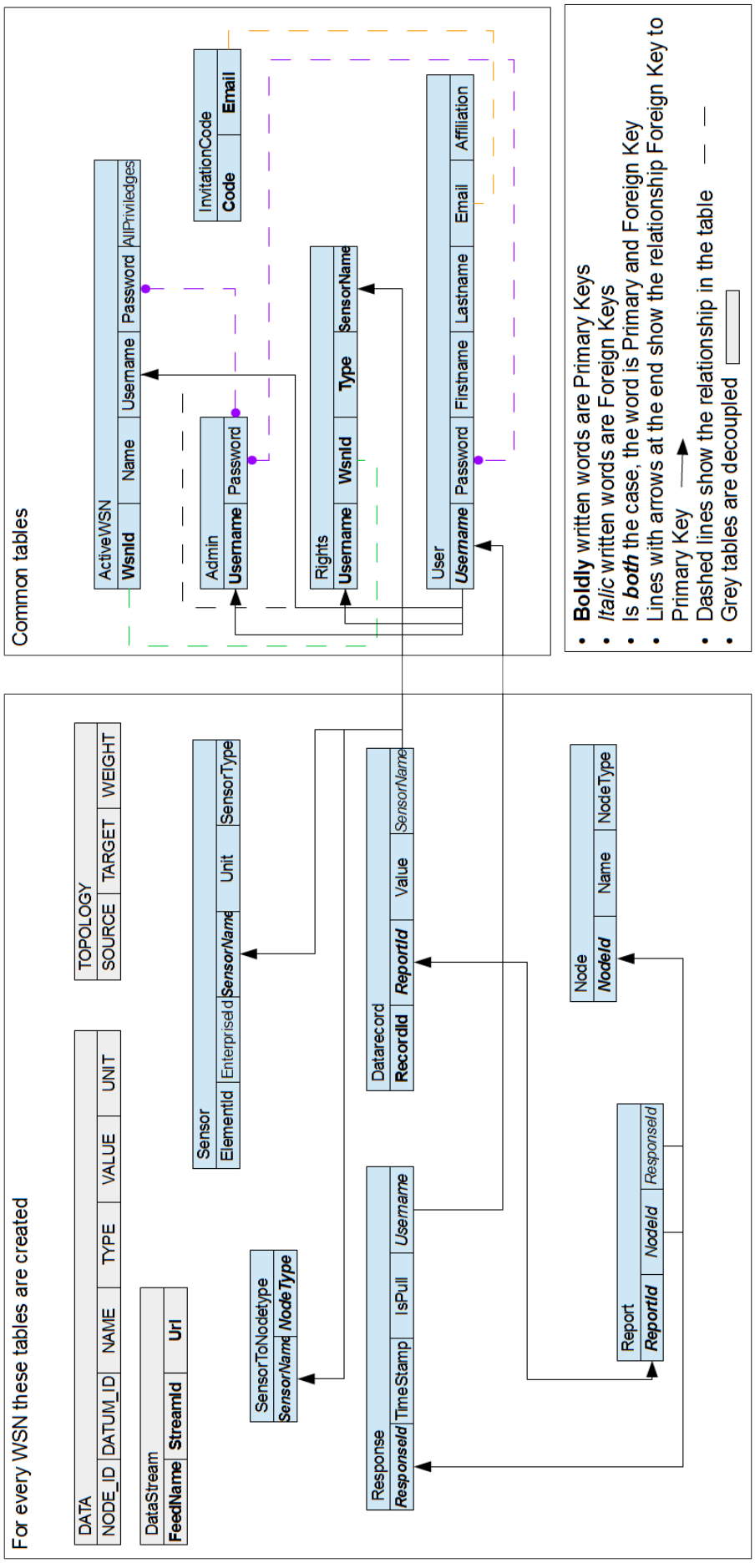

Figure 2.2: WebMaDa database schema

Chapter 3

Design Decisions

This chapter shows the design decisions that were made and why they were made. The main parts here are the filter options which were chosen and how this data can be printed or be saved for later. The whole process of login, choosing a WSN and filter data is shown in Figure 3.1.
Figure 3.1 has two main color themes. The blue bits are the path the user has to make, like a navigation through the process. The green boxes show what is going on behind this process.
The whole process starts with login in to WebMaDa. The login might be successful or not however for this work it is only important if the user can log in. Through the navigation steps listed in Figure 3.1 the user can navigate to the WSNs on which he/she is authorized to push/pull. If the user chooses a WSN where he/she is not the owner of, it is not relevant for this work. If the user is the owner of the chosen WSN he/she will see a new option: Filter Options. In this option the user first has to select a username, a date range and if it only should show results with push or pull or both. After the button with "Show filtered data" is pressed, the user will see the filtered data of the filter settings from before.
Concerning the green squares the first one is when the page is loaded. It will get all the usernames of the user that are authorized to push/pull in the chosen WSN. The second block shows the query to the database for the date range selection buttons. The query result will be a date of the first and last action in the chosen WSN. When the button "Show filtered data" is hit the query of the fourth block will be executed. The interesting part in this query are the different cases listed and the table in the lower right corner. Depending on the chosen filter settings this query will be different. If "All users" is selected by the user the query stays the same because we do not filter for a special user. Is one or more user/s chosen the "Specific user" case will be added for the right amount of chosen users. If "Push and Pull" is chosen we have the same case as for "All users". Is only push or pull chosen the query will be extended accordingly with IsPull = False/True. In the end this result will be sent back to WebMaDa and displayed in a table which can be sorted, printed and saved as a pdf.

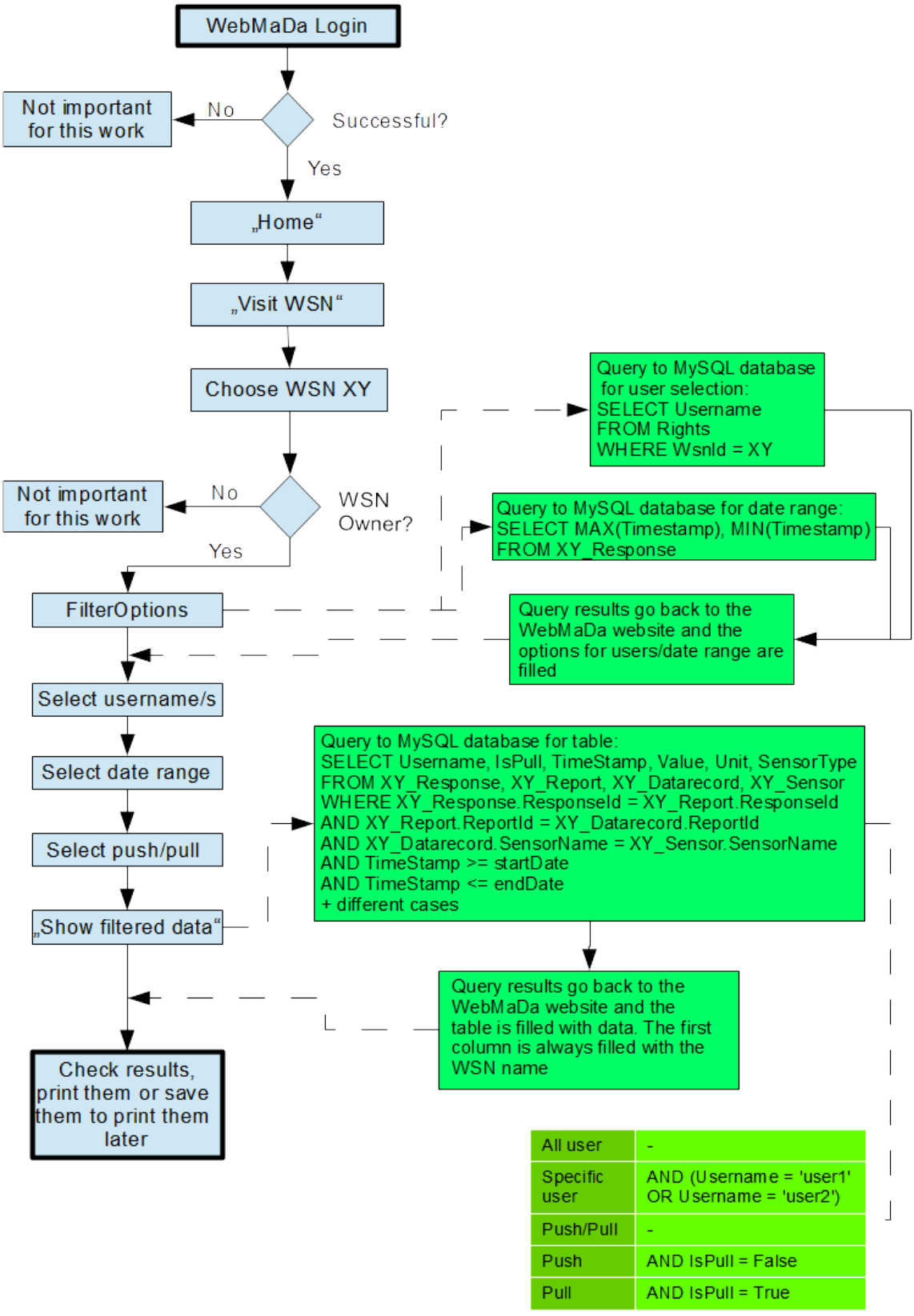

Figure 3.1: Transparency flow

3.1 Filtering options

The main goal of this work is to give the owner of a WSN more transparency of his WSN. The important information that a WSN owner can get from this are who pulled or pushed what and when. Therefore these options where chosen as filter options:

- User
- Push/Pull
- Date
- Data

For a better understanding a mockup of the website with the filtering options and the resulting table was created (see Figure 3.2).

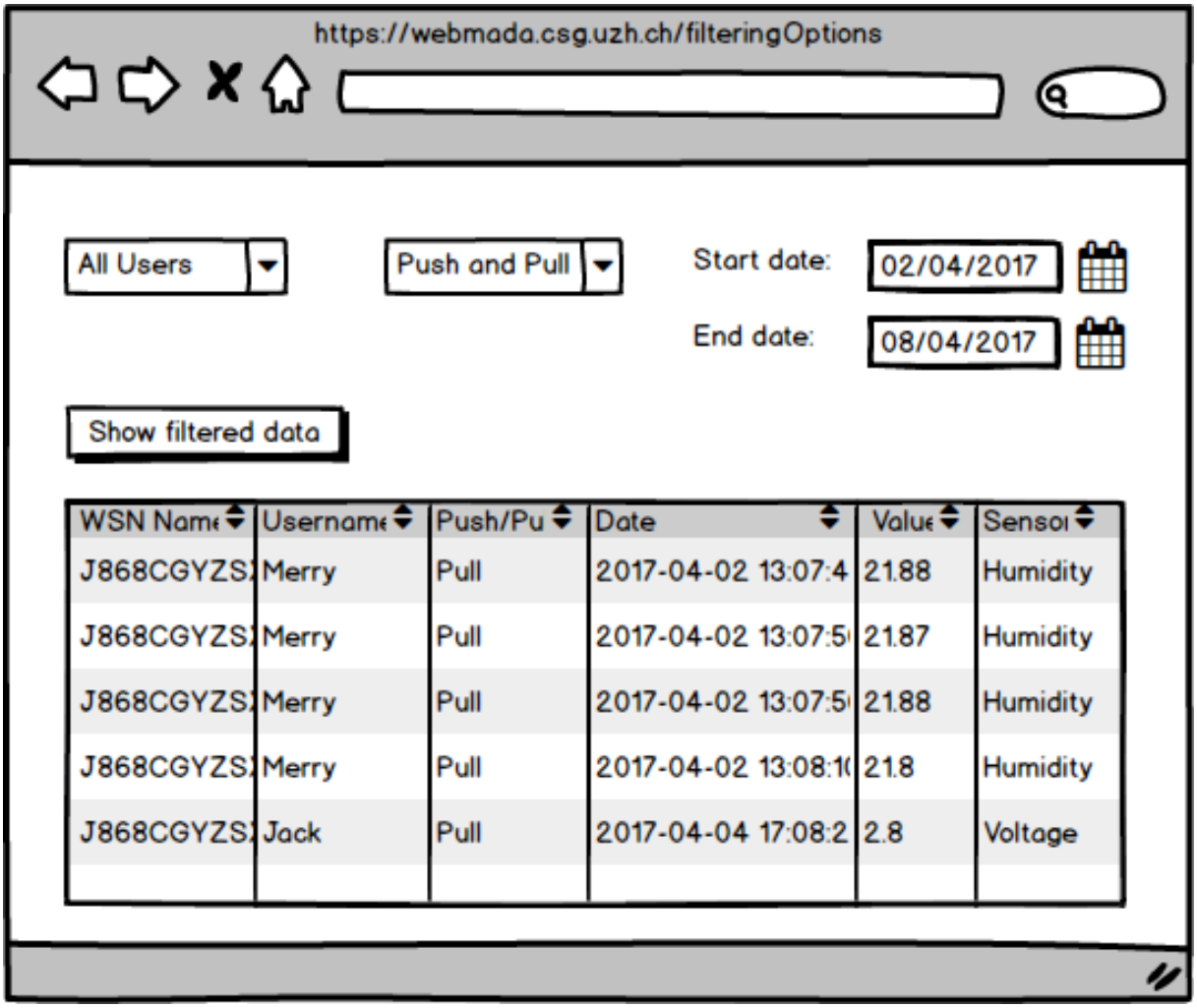

Figure 3.2: Filtered table mockup

The first column has to be the WSN name because an owner can have multiple WSNs and if he/she wants to compare data from different WSNs the name must be present in this table. The username of the user, who made a request is also very important because when this information would be missing looking at this data would be a lot more meaningless. If the request was a push or a pull request is also fundamental. The date is important because one might wants to know when data was pushed if something went wrong. The last two columns show the value and what this value means. This is for showing what the person, who made the request was interested in.

3.2 Filtered Data

After the table was filled with the requested filtered data, the user can also choose a column on which the data should be sorted. For example the data could sorted by date which would end in either the first date being the first entry or the first date being the last entry. When the user is satisfied with the table it should be possible to either print this table directly in the browser or save the table to a folder on the desktop. The file will be named like WSNName_Date_Time. That way it is easy to sort the files if there are more than one. The sorted files are sorted in WSN names and after that it is sorted after date and time. This way a chronological order is provided. The header of the files will also include WSN name, date and time because if the tables are printed they can still be easily ordered.

Chapter 4

Implementation

This chapter is about how the idea of the work and the design decisions were implemented in the live-system of WebMaDa. The structure is like the work process. First will be shown how a new page and the GUI elements were made, filled through the database and then read out again. After that the query to get the filtered data will be discussed. The next point will show how the table is filled and how the table is made sortable and finally the last part will be how the table is saved in a PDF.

4.1 Pre work

Before anything is displayed we are in the filtering.php file. There it is checked if you are the owner of the WSN or not. Additionally several parameters like start date, end date and WsnId are stored.

```
105  // Get the time range a wsn existed
106  function getTimeRange($wsnId){
107      $handle = Config::$pdo->prepare('CALL GetTimeRange(?)');
108      $handle->bindValue(1, $wsnId);
109      $handle->execute();
110
111      $result = $handle->fetchAll();
112      $handle->closeCursor();
113      return $result;
114  }
```

Figure 4.1: Function in filtering.php

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `GetTimeRange`(IN p_wsnid VARCHAR(30))
BEGIN
SET @s = CONCAT('SELECT date(min(TimeStamp)) as sDate, date(max(TimeStamp)) as eDate FROM  ', p_wsnid, '_Response');
PREPARE stmt FROM @s;
EXECUTE stmt;
DEALLOCATE PREPARE stmt;
END
```

Figure 4.2: Stored procedure to Figure 4.1

4.2 Adding a new option

As Filter Options were an addition and not an extension a new page only for the Filter Options was needed. This was achieved by adding a button in the navigation.tpl file as seen in Figure 4.1.

```
<!-- This is the Smarty layout for the navigation panel in the WSN views-->
<div class="row">
    <div class="col-md-2 textmiddle">
        <a class="list-group-item {if $active === 'wsn-data'} active {/if}" href="wsn-data.php?wsn={$wsnId}">WSN data</a>
    </div>

    {if isset($isOwner) && $isOwner == true}
    <!--<div class="col-md-3 textmiddle">
        <a class="list-group-item {if $active === 'topology'} active {/if}" href="visualizeTopology.php?wsn={$wsnId}">Legacy: Topology</a>
    </div>-->
    <div class="col-md-3 textmiddle">
        <a class="list-group-item {if $active === 'streams'} active {/if}" href="visualizeDatastream.php?wsn={$wsnId}">Legacy: Datastreams</a>
    </div>
    <div class="col-md-2 textmiddle">
        <a class="list-group-item {if $active === 'raw'} active {/if}" href="visualizePackages.php?wsn={$wsnId}">Legacy: Raw data</a>
    </div>
    <div class="col-md-2 textmiddle">
        <a class="list-group-item {if $active === 'my-wsns'} active {/if}" href="my-wsns.php?wsn={$wsnId}">WSN administration</a>
    </div>
    <div class="col-md-2 textmiddle">
        <a class="list-group-item {if $active === 'filtering'} active {/if}" href="filtering.php?wsn={$wsnId}">Filter Options</a>
    </div>
    {else}
    <div class="col-md-2 textmiddle">
        <a class="list-group-item {if $active === 'other-wsns'} active {/if}" href="other-wsns.php">View privileges</a>
    </div>
    {/if}
</div>

```

Figure 4.3: navigation.tpl with new button

In Figure 4.2 the result of the above written code in navigation.tpl can be seen. The new button ”Filter Options” is added on the right-hand side.

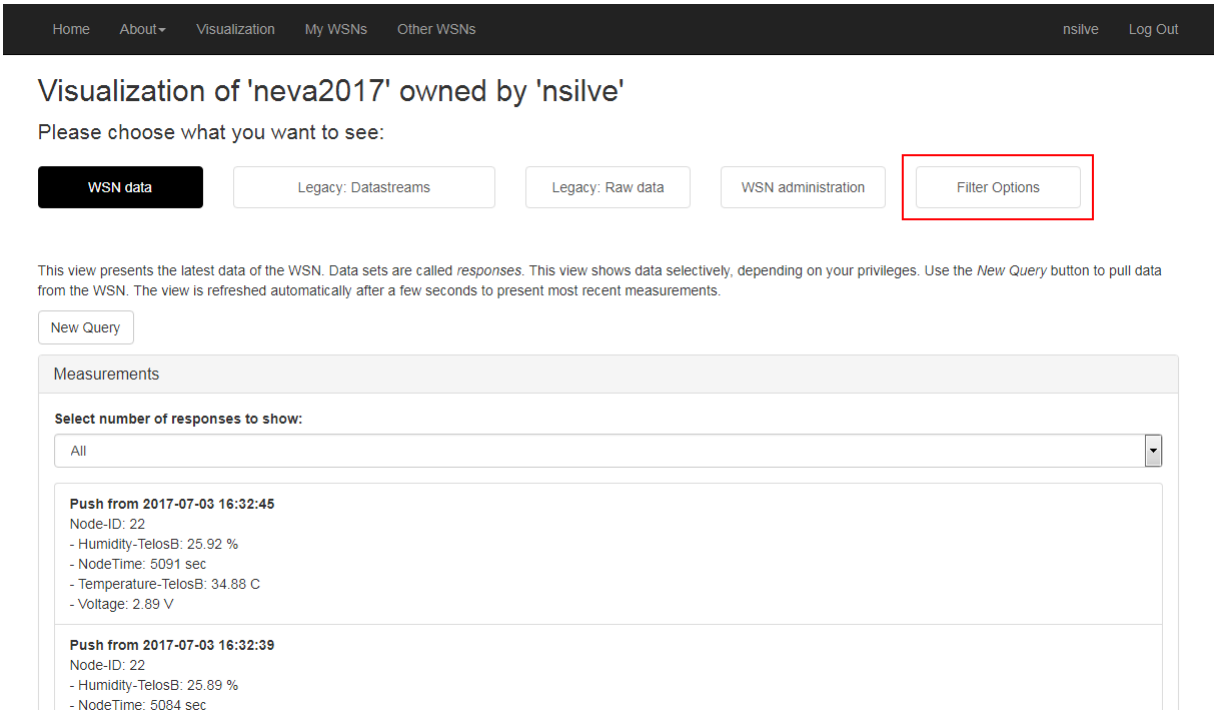

Figure 4.4: Filter Options button

4.3 GUI elements

In chapter 3 Design Decisions four options were chosen to be the options for filtering data. These were:

- User
- Push/Pull
- Date
- Data

User: The WSN owner could possibly search for one single user or multiple specific user but also for all users. Therefore the best solution to pick these options was a list where the WSN owner can choose multiple users or just ”All users”. This list was created in a new template file called filtering.tpl. This file resides with the other layout files in the folder ”layout”. Figure 4.3 shows the code and Figure 4.4 show the result in WebMaDa.

In the same manner also a drop-down menu for the option to choose between pull or push or both was added. Two fields were added for start and end date. Start and end date can only be chosen in the range where the WSN existed. Also two buttons ”Show filtered data” and ”Save filtered data” were added to the view. Concluded this resulted in the view seen in Figure 4.5.
Apart from the visible elements on the page there are also two hidden elements. The first

```
30      <div class="panel-body" id="own-wsns-div">
31          <div id="user-select-div">
32              <label>Select one or more user/s</label>
33              <select multiple size="3" id="user-select" class="form-control">
34                  <option value="allUsers" name="allUsers" selected>
35                      All Users
36                  </option>';
37                  {foreach $ownWsns as $wsn}
38                      {foreach $wsn.Users as $user}
39                          {if $wsn.WsnId === $wsnId}
40                              <option value="{$user.Username}" name="{$user.Username}" selected>
41                                  {$user.Username}
42                              </option>';
43                          {/if}
44                      {/foreach}
45                  {/foreach}
46              </select>
47          </div>
```

Figure 4.5: Users GUI element

Figure 4.6: Users code for GUI element

one is the text which is displayed when no data can be retrieved from a filtering. The second hidden element is the table where the filtered data will be displayed in. This table contains only the header at this point but will be filled dynamically later.

Figure 4.7: Finished filter options

4.4 The query and the table

In Figure 3.1 the next step is the huge query to the MySQL database. But before that the filter options fields have to be read and stored.

```
SELECT Username, IsPull, TimeStamp, Value, Unit, SensorType
FROM XY_Response, XY_Report, XY_Datarecord, XY_Sensor
```

WHERE XY_Response.ResponseId = XY_Report.ResponseId
AND XY_Report.ReportId = XY_Datarecord.ReportId
AND XY_Datarecord.SensorName = XY_Sensor.SensorName
AND TimeStamp >= startDate
AND TimeStamp <= endDate
+ different cases

In Figure 4.6 is the whole procedure from reading and storing the variables to sending the data to a php file, where the connection to the database is establish, until getting back the data from the php file and visualize it.
Lines 125 - 129 read and store the filter options. Line 131 describes the data container that is sent via the ajax call. (Note: ldelim and rdelim describes a normal "" and "" but because smarty uses these parentheses for passing variables they can't be used with normal text.).
Lines 135 - 137 describe the ajax call setting. It is a POST call because we want to submit the data from the filtering options. The url describes the path where this ajax call should

```
124 $("#showBtn").click(function(){
125     var users = $('#user-select').val();
126     var e = document.getElementById("pushPull-select");
127     var pushPull = e.options[e.selectedIndex].text;
128     var startDate = document.getElementById("startDate").value;
129     var endDate = document.getElementById("endDate").value;
130 
131     var parameters = {ldelim}WsnId: '{$wsnId}', Users: users, PushPull: pushPull,
132     StartDate: startDate, EndDate: endDate {rdelim};
133 
134    $.ajax({
135       type: 'POST',
136       url: 'filtering_gathering.php',
137       data: parameters,
138       success: function(result) {
139                 if(result.length == 2){
140                     document.getElementById("filteredData").style.display = "none";
141                     document.getElementById("noResults").style.visibility = "visible";
142                 }
143                 else{
144                     document.getElementById("noResults").style.visibility = "hidden";
145                     $("#filteredData tr:gt(0)").remove();
146                     document.getElementById("filteredData").style.display = "";
147 
148                     var trHTML = '<tbody>';
149                     $.each(JSON.parse(result), function (i, item) {
150                         trHTML += '<tr><td class="body" align="center" valign="middle">' +
151                         "{$wsnId}" + '</td><td class="body" align="center" valign="middle">' +
152                         item.Username + '</td><td class="body" align="center" valign="middle">' +
153                         item.IsPull + '</td><td class="body" align="center" valign="middle">' +
154                         item.TimeStamp + '</td><td class="body" align="center" valign="middle">' +
155                         item.Value + '</td><td class="body" align="center" valign="middle">' +
156                         item.Unit + '</td><td class="body" align="center" valign="middle">' +
157                         item.SensorType + '</td></tr>';
158                     });
159                     trHTML += '</tbody> ';
160                     $('#filteredData').append(trHTML);
161                     $("#filteredData").DataTable();
162                 }
163         }
164    });
```

Figure 4.8: Show filtered data submit button

go to, here it is another php file. The data that is ”posted” is the container parameters. Here the ajax call is proceeding to the php file called filtering-gathering.php.
The file filtering-gathering.php is a very simple file. It has two main parts. The first part is where the POST variables are checked and then they are saved. Then a function called ”GetFilteredData” is called. This function is as it suggests by its name to get filtered data from the database. In this function we have four cases:

- Case 1: ”All users” and ”Push and Pull”
- Case 2: ”All users” and ”Push” or ”Pull”
- Case 3: Any user/s and ”Push and Pull”
- Case 4: Any user/s and ”Push” or ”Pull”

In MySQL it is not recommended to work with case statements, therefore for every case a stored procedure was made.

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `GetFilteredDataAllUserPP`(IN p_wsnid VARCHAR(30),
IN p_sDate date, IN p_eDate date)
BEGIN
SET @s = CONCAT(
'SELECT Username, IsPull, TimeStamp, Value, Unit, SensorType
FROM  ',p_wsnid, '_Response, ',p_wsnid, '_Report, ',p_wsnid, '_Datarecord, ',p_wsnid, '_Sensor
WHERE ',p_wsnid, '_Response.ResponseId = ',p_wsnid, '_Report.ResponseId
AND ',p_wsnid, '_Report.ReportId = ',p_wsnid, '_Datarecord.ReportId
AND ',p_wsnid, '_Datarecord.SensorName = ',p_wsnid, '_Sensor.SensorName
AND date(TimeStamp) >= "',p_sDate,'"
AND date(TimeStamp) <= "',p_eDate,'"
');
PREPARE stmt FROM @s;
EXECUTE stmt;
DEALLOCATE PREPARE stmt;
END
```

Figure 4.9: Case 1 stored procedure

The Case 1 stored procedure looks like the base query because for ”All Users” and ”Push and Pull” you don’t have to set an extra field.

The Case 2 stored procedure has the additional field for ”IsPull” because in this case ”Push and Pull” cannot be selected and therefore we have to filter for either push or pull.

The Case 3 stored procedure doesn’t have the ”IsPull” field but one for the users. The array of user that we get from the ajax call is prepared in the filtering-gathering.php file to fit in a database query.

The Case 4 stored procedure is so to speak the normal case where not any special fields like ”All Users” or ”Push and Pull” are chosen and therefore for every filter option there is a field in the query.
When the query can be executed it will get back a table with the filtered data. This table is returned to the ajax call under the success field on line 138. After that the array ”result” is checked for its length to see if it is empty which means that the filter settings gave no

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `GetFilteredDataAllUser`(IN p_wsnid VARCHAR(30),
 IN p_pushPull VARCHAR(30), IN p_sDate date, IN p_eDate date)
BEGIN
SET @s = CONCAT(
'SELECT Username, IsPull, TimeStamp, Value, Unit, SensorType
FROM  ',p_wsnid, '_Response, ',p_wsnid, '_Report, ',p_wsnid, '_Datarecord, ',p_wsnid, '_Sensor
WHERE ',p_wsnid, '_Response.ResponseId = ',p_wsnid, '_Report.ResponseId
AND ',p_wsnid, '_Report.ReportId = ',p_wsnid, '_Datarecord.ReportId
AND ',p_wsnid, '_Datarecord.SensorName = ',p_wsnid, '_Sensor.SensorName
AND IsPull = ',p_pushPull,'
AND date(TimeStamp) >= "',p_sDate,'"
AND date(TimeStamp) <= "',p_eDate,'"
');
PREPARE stmt FROM @s;
EXECUTE stmt;
DEALLOCATE PREPARE stmt;
END
```

Figure 4.10: Case 2 stored procedure

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `GetFilteredDataPP`(IN p_wsnid VARCHAR(30),
 IN p_users VARCHAR(250), IN p_sDate date, IN p_eDate date)
BEGIN
SET @s = CONCAT(
'SELECT Username, IsPull, TimeStamp, Value, Unit, SensorType
FROM  ',p_wsnid, '_Response, ',p_wsnid, '_Report, ',p_wsnid, '_Datarecord, ',p_wsnid, '_Sensor
WHERE ',p_wsnid, '_Response.ResponseId = ',p_wsnid, '_Report.ResponseId
AND ',p_wsnid, '_Report.ReportId = ',p_wsnid, '_Datarecord.ReportId
AND ',p_wsnid, '_Datarecord.SensorName = ',p_wsnid, '_Sensor.SensorName
AND ',p_users,'
AND date(TimeStamp) >= "',p_sDate,'"
AND date(TimeStamp) <= "',p_eDate,'"
');
PREPARE stmt FROM @s;
EXECUTE stmt;
DEALLOCATE PREPARE stmt;
END
```

Figure 4.11: Case 3 stored procedure

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `GetFilteredData`(IN p_wsnid VARCHAR(30),
 IN p_users VARCHAR(250), IN p_pushPull VARCHAR(30), IN p_sDate date, IN p_eDate date)
BEGIN
SET @s = CONCAT(
'SELECT Username, IsPull, TimeStamp, Value, Unit, SensorType
FROM  ',p_wsnid, '_Response, ',p_wsnid, '_Report, ',p_wsnid, '_Datarecord, ',p_wsnid, '_Sensor
WHERE ',p_wsnid, '_Response.ResponseId = ',p_wsnid, '_Report.ResponseId
AND ',p_wsnid, '_Report.ReportId = ',p_wsnid, '_Datarecord.ReportId
AND ',p_wsnid, '_Datarecord.SensorName = ',p_wsnid, '_Sensor.SensorName
AND ',p_users,'
AND IsPull = ',p_pushPull,'
AND date(TimeStamp) >= "',p_sDate,'"
AND date(TimeStamp) <= "',p_eDate,'"
');
PREPARE stmt FROM @s;
EXECUTE stmt;
DEALLOCATE PREPARE stmt;
END
```

Figure 4.12: Case 4 stored procedure

results. This can be the case if a person in the WSN has pull rights but never made a pull and the filter options were specific for that person with the option pull. If that is the case a short message will be visible that with these settings no data is available.
If ”result” is not empty the table that may already exists, is cleared and made visible. After that an html string is built to dynamically load the html table. The newly created string is appended to the table and the table is loaded as DataTable().

4.4.1 DataTables

DataTables [7] is a plug-in for the jQuery Javascript library. With this plug-in tables can be easily transformed to sortable and nice looking tables. It also features a paging functionality if the data set of the table is very large and a search bar to search the table.

4.5 HTML/Javascript to PDF

As seen in Figure 4.5 there is a second button next to the ”Show filtered data” which is called ”Save filtered data”. This button is used to save the current view of the table to a pdf. This is achieved with the plug-in jsPDF [8].

Line 169 - 173 is as before to read and store the filter option values. These are needed again to write them to the header of the file. Line 175 initializes a new pdf. Lines 176 and 177 are to get the current date and time to write it to the file name later. Lines 184 -192 show how the header is written. The next line is how the html table is read. Finally the last line is the saving of the file with the name structure of WsdId followed by date and time.

```
168 $("#saveBtn").click(function(){
169     var users = $('#user-select').val();
170     var e = document.getElementById("pushPull-select");
171     var pushPull = e.options[e.selectedIndex].text;
172     var startDate = document.getElementById("startDate").value;
173     var endDate = document.getElementById("endDate").value;
174 
175     var pdf = new jsPDF('l','pt','a3');
176     var date = new Date().toLocaleDateString();
177     var time = new Date().toLocaleTimeString();
178     margins = {
179       top: 120,
180       bottom: 60,
181       left: 40,
182       width: 400
183     };
184     pdf.setFontSize(15);
185     pdf.setFontType("bold");
186     pdf.text(40, 30, 'Used options:');
187     pdf.setFontSize(13);
188     pdf.setFontType("normal");
189     pdf.text(40, 60, 'Users: ' + users);
190     pdf.text(40, 80, 'Push/Pull: ' + pushPull);
191     pdf.text(40, 100, 'Start date: ' + startDate);
192     pdf.text(190, 100, 'End date: ' + endDate   );
193 
194     pdf.addHTML(document.getElementById('filteredData'),
195     margins.left, margins.top,{ldelim}pagesplit:true{rdelim},function() {
196         pdf.save('{$wsnId}_'+ date +'_' + time +'.pdf');
197     });
198 });
```

Figure 4.13: Save data submit button

Chapter 5

Evaluation

The mean to proof that my work is correct will be a Proof of Operability. This is the case because this is a web-based program, which means that it has hardly any limits in resources and therefore measuring f.e. disk space would be useless.
Followed are a set of screenshot which show that the chosen filter options yield the correct result.

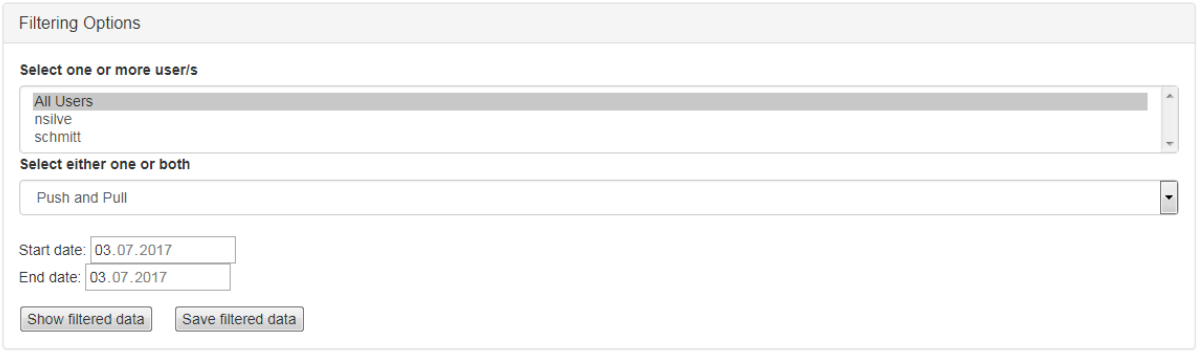

Figure 5.1: Filter option F1

The filter options are: ”All Users” and ”Push and Pull”. As in Figure 5.2 can be seen all authorized users are on the table with push and pull requests.

Show 100 entries Search:

Wsn Name	Username	IsPull	TimeStamp	Value	Unit	SensorType
SJLQVKI5RG	schmitt	1	2017-07-03 15:18:46	651	sec	NodeTime
SJLQVKI5RG	schmitt	1	2017-07-03 15:35:39	1664	sec	NodeTime
SJLQVKI5RG	schmitt	1	2017-07-03 15:18:22	33.82	C	Temperature
SJLQVKI5RG	schmitt	1	2017-07-03 15:30:45	34.55	C	Temperature
SJLQVKI5RG	schmitt	1	2017-07-03 15:36:13	34.68	C	Temperature
SJLQVKI5RG	schmitt	1	2017-07-03 15:18:46	33.84	C	Temperature
SJLQVKI5RG	schmitt	1	2017-07-03 15:35:39	34.66	C	Temperature
SJLQVKI5RG	schmitt	1	2017-07-03 15:22:49	2.89	V	Voltage
SJLQVKI5RG	schmitt	1	2017-07-03 15:18:46	2.89	V	Voltage
SJLQVKI5RG	schmitt	1	2017-07-03 15:35:39	2.89	V	Voltage
SJLQVKI5RG	nsilve	1	2017-07-03 15:42:47	27.4	%	Humidity
SJLQVKI5RG	nsilve	0	2017-07-03 15:15:52	29.61	%	Humidity
SJLQVKI5RG	nsilve	0	2017-07-03 15:16:53	29.26	%	Humidity
SJLQVKI5RG	nsilve	0	2017-07-03 15:17:00	29.19	%	Humidity
SJLQVKI5RG	nsilve	0	2017-07-03 15:17:07	29.15	%	Humidity
SJLQVKI5RG	nsilve	0	2017-07-03 15:17:13	29.12	%	Humidity
SJLQVKI5RG	nsilve	0	2017-07-03 15:17:20	29.12	%	Humidity
SJLQVKI5RG	nsilve	0	2017-07-03 15:17:27	29.04	%	Humidity
SJLQVKI5RG	nsilve	0	2017-07-03 15:17:34	29.01	%	Humidity

Figure 5.2: Filter options result R1

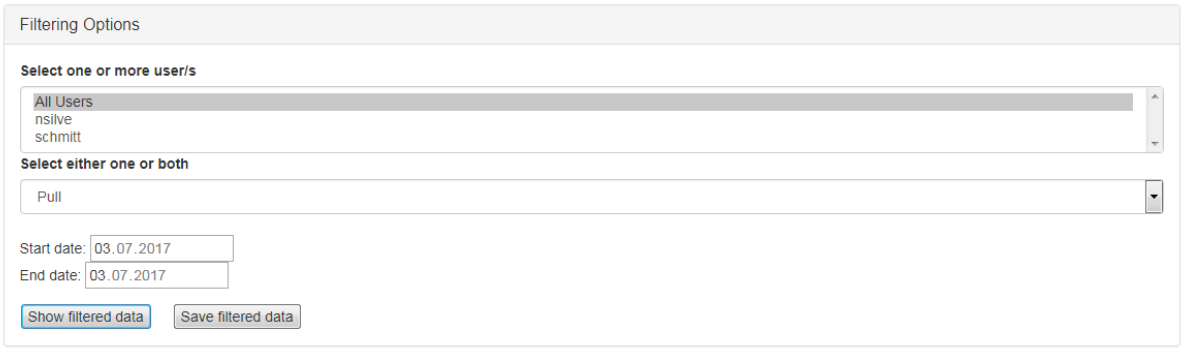

Figure 5.3: Filter option F2

Show 10 entries Search:

Wsn Name	Username	IsPull	TimeStamp	Value	Unit	SensorType
SJLQVKI5RG	nsilve	1	2017-07-03 15:42:47	27.4	%	Humidity
SJLQVKI5RG	schmitt	1	2017-07-03 15:18:46	651	sec	NodeTime
SJLQVKI5RG	schmitt	1	2017-07-03 15:35:39	1664	sec	NodeTime
SJLQVKI5RG	nsilve	1	2017-07-03 15:42:47	2093	sec	NodeTime
SJLQVKI5RG	nsilve	1	2017-07-03 15:42:59	2104	sec	NodeTime
SJLQVKI5RG	schmitt	1	2017-07-03 15:18:22	33.82	C	Temperature
SJLQVKI5RG	schmitt	1	2017-07-03 15:30:45	34.55	C	Temperature
SJLQVKI5RG	schmitt	1	2017-07-03 15:36:13	34.68	C	Temperature
SJLQVKI5RG	schmitt	1	2017-07-03 15:18:46	33.84	C	Temperature
SJLQVKI5RG	schmitt	1	2017-07-03 15:35:39	34.66	C	Temperature

Figure 5.4: Filter option result R2

The filter options are: ”All Users” and ”Pull”. As in Figure 5.4 can be seen all authorized users are on the table with only pull requests.

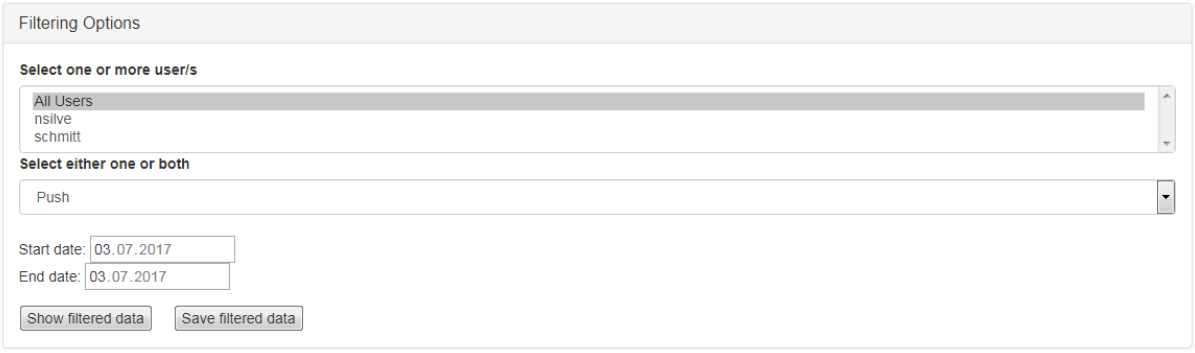

Figure 5.5: Filter option F3

Show 10 entries Search:

Wsn Name	Username	IsPull	TimeStamp	Value	Unit	SensorType
SJLQVKI5RG	nsilve	0	2017-07-03 15:15:52	29.61	%	Humidity
SJLQVKI5RG	nsilve	0	2017-07-03 15:16:53	29.26	%	Humidity
SJLQVKI5RG	nsilve	0	2017-07-03 15:17:00	29.19	%	Humidity
SJLQVKI5RG	nsilve	0	2017-07-03 15:17:07	29.15	%	Humidity
SJLQVKI5RG	nsilve	0	2017-07-03 15:17:13	29.12	%	Humidity
SJLQVKI5RG	nsilve	0	2017-07-03 15:17:20	29.12	%	Humidity
SJLQVKI5RG	nsilve	0	2017-07-03 15:17:27	29.04	%	Humidity
SJLQVKI5RG	nsilve	0	2017-07-03 15:17:34	29.01	%	Humidity
SJLQVKI5RG	nsilve	0	2017-07-03 15:17:41	28.97	%	Humidity
SJLQVKI5RG	nsilve	0	2017-07-03 15:17:48	28.94	%	Humidity

Figure 5.6: Filter option result R3

The filter options are: ”All Users” and ”Push”. As in Figure 5.6 can be seen one authorized user is on the table with only push requests.

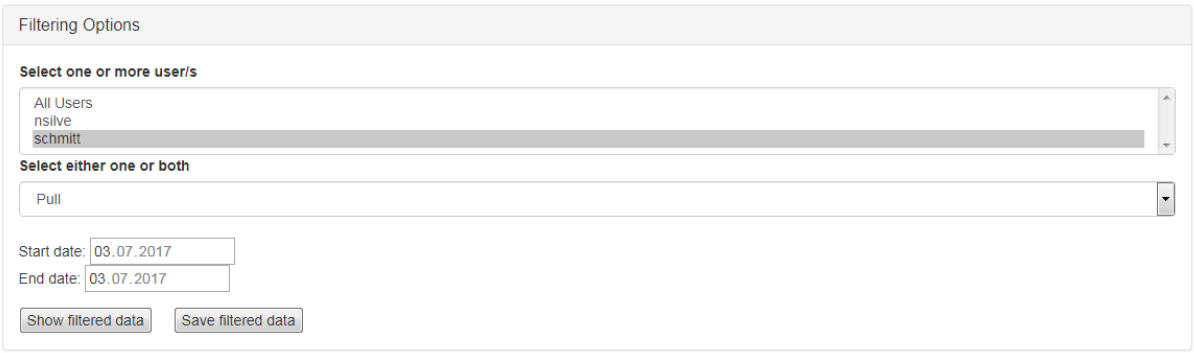

Figure 5.7: Filter option F4

Show 10 entries Search:

Wsn Name	Username	IsPull	TimeStamp	Value	Unit	SensorType
SJLQVKI5RG	schmitt	1	2017-07-03 15:18:46	651	sec	NodeTime
SJLQVKI5RG	schmitt	1	2017-07-03 15:35:39	1664	sec	NodeTime
SJLQVKI5RG	schmitt	1	2017-07-03 15:18:22	33.82	C	Temperature
SJLQVKI5RG	schmitt	1	2017-07-03 15:18:46	33.84	C	Temperature
SJLQVKI5RG	schmitt	1	2017-07-03 15:30:45	34.55	C	Temperature
SJLQVKI5RG	schmitt	1	2017-07-03 15:35:39	34.66	C	Temperature
SJLQVKI5RG	schmitt	1	2017-07-03 15:36:13	34.68	C	Temperature
SJLQVKI5RG	schmitt	1	2017-07-03 15:18:46	2.89	V	Voltage
SJLQVKI5RG	schmitt	1	2017-07-03 15:22:49	2.89	V	Voltage
SJLQVKI5RG	schmitt	1	2017-07-03 15:35:39	2.89	V	Voltage

Figure 5.8: Filter option result R4

The filter options are: ”schmitt” and ”Pull”. As in Figure 5.8 can be seen only the chosen authorized user is on the table with only pull requests.

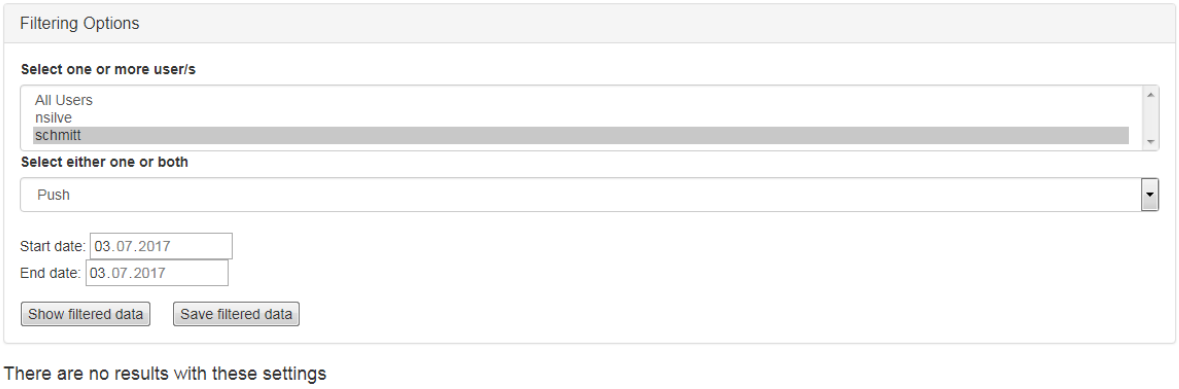

Figure 5.9: Filter option F5 and filter option result R1

The filter options are: ”schmitt” and ”Pull”. As in Figure 5.9 can be seen there are no results for this query, which is true.

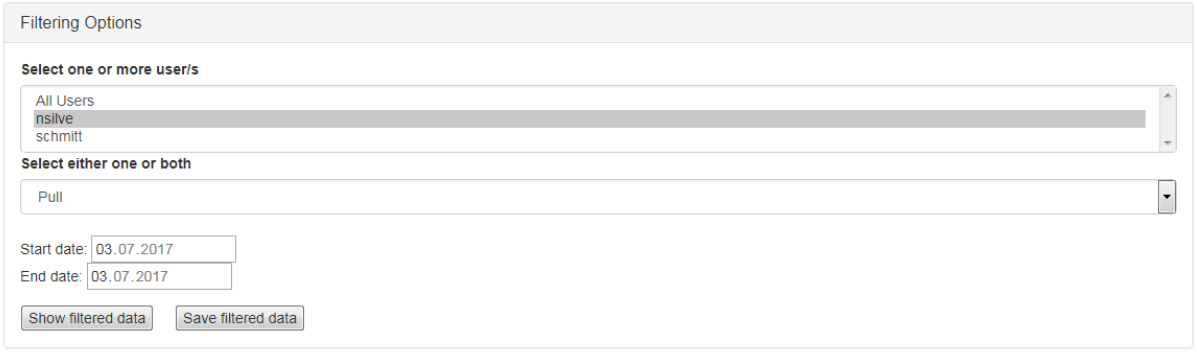

Figure 5.10: Filter option F6

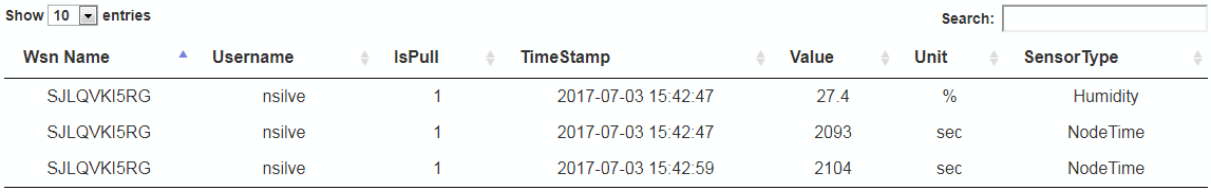

Figure 5.11: Filter option result R6

The filter options are: ”nsilve” and ”Pull”. As in Figure 5.11 can be seen only the chosen authorized user is on the table with only pull requests.
All tests yielded the correct result, which concludes my proof of operability.

Chapter 6

Summary and Conclusions

To WebMaDa an additional option for WSN owner in their own WSN was added. It is called ”Filter Options” and was added as the last item of the list which only WSN owner can see when they choose one of their WSNs. In this addition a WSN owner can choose between users that are eligible for any action in their WSN or he can choose all of them. It is possible to choose between ”Push and Pull”, ”Push” and ”Pull” which makes it easier if the WSN owner is only interested in one type of data. The last option to choose from is date. It is only possible to choose a date between the dates when the WSN had its first and last pull. If all options are chosen, the WSN owner can press the button ”Show filtered data” which will then generate a table with the filtered data. On a first appearance the table can be sorted by any of the columns. If the WSN Owner would like to keep the table he can press the other available button, which is called ”Save filtered data”. The PDF file will be generated automatically and saved to the downloads folder of the browser.

The goals of the thesis where to upgrade the existing WebMaDa with a user-friendly GUI with different filtering options to see who accessed what data when. This goal was fulfilled with the new option in the interface called ”Filter Options”. As stated in the above section there are different filtering options of which can be seen who accessed what data when. The second goal was to design secure database access and receive specific data answering the request from data owners and the implementation of it. The secure database access was developed in PHP files according to other similar files in WebMaDa. Through the PHP files five new stored procedures are called and executed which fulfils this goal. The fourth goal was the evaluation. In this report the evaluation was given through a Proof of Operability. In one WSN all the different test cases were looked at. The resulting tables were all as the expected result which concludes the Proof of Operability.

Bibliography

[1] H. Karl and A. Willig, *Protocols and Architecture for Wireless Sensor Networks*, John Wiley & Sons, Vol. 1, ISBN: 0470519231, GB, 2007.

[2] SecureWSN, `http://www.csg.uzh.ch/csg/en/research/SecureWSN.html`, Last visited 18.06.2017.

[3] C. Schmitt, T. Strasser, B. Stiller: *Third-party-independent Data Visualization of Sensor Data in CoMaDa*; 12th IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, New York, U.S.A., Oct. 2016, pp 1-8.

[4] C.Schmitt, C.Anliker, B.Stiller: Efficient and Secure Pull Requests for Emergency Cases Using a Mobile Access Framework; Managing the Web of Things: Linking the Real World to the Web, M.Sheng, Y. Qin, L. Yao, and B. Benatallah (Eds.), Morgen Kaufmann (imprint of Elsevier), Chapter 8, pp. 229-247, ISBN: 978-0-12-809764-9, February 2017.

[5] André Freitag, Corinna Schmitt, Georg Carle: CoMaDa: *An Adaptive Framework with Graphical Support for Configuration*; 9th International Conference on Network and Service Management, Zürich, Switzerland, October 2013, ISBN 978-3-901882-53-1, pp 211-218.

[6] Communication Systems Group (CSG), `http://www.csg.uzh.ch`, Last visited 18.06.2017.

[7] DataTables, `https://datatables.net`, Last visited 03.07.2017.

[8] jsPdf, `https://github.com/MrRio/jsPDF`, Last visited 03.07.2017.

[9] MySQL, `https://dev.mysql.com/doc/connector-net/en/connector-net-tutorials-stored-procedures.html`, Last visited 03.07.2017.

[10] Smarty, `http://www.smarty.net/docs/en/language.function.html.checkboxes.tpl`, Last visited 03.07.2017.

[11] stackoverflow, `https://stackoverflow.com/questions/8984208/using-smarty-templates-to-send-emails`, Last visited 03.07.2017.

[12] Smarty, `http://www.smarty.net/forums/viewtopic.php?p=88222&sid=0682759a5051ba6ec5887e4285fbde39`, Last visited 03.07.2017.

[13] stackoverflow, https://stackoverflow.com/questions/1085801/get-selected-value-in-dropdown-list-using-javascript, Last visited 03.07.2017.

[14] Bigace, http://www.bigace.de/Smarty-Open-Source-Template-Engine-for-PHP-Part-1.html, Last visited 03.07.2017.

[15] stackoverflow, https://stackoverflow.com/questions/22217635/get-json-object-from-ajax-call, Last visited 03.07.2017.

[16] stackoverflow, https://stackoverflow.com/questions/6323338/jquery-ajax-posting-json-to-webservice, Last visited 03.07.2017.

[17] jQuery, https://forum.jquery.com/topic/syntax-error-on-ajax-load-post-object, Last visited 03.07.2017.

[18] Asprunner, http://www.asprunner.com/forums/topic/6731-passing-smarty-variable-to-javascript-function/, Last visited 03.07.2017.

[19] stackoverflow, https://stackoverflow.com/questions/18680248/html-button-onclick-event, Last visited 03.07.2017.

[20] stackoverflow, https://stackoverflow.com/questions/14976495/get-selected-option-text-with-javascript, Last visited 03.07.2017.

[21] stackoverflow, https://stackoverflow.com/questions/3002165/echo-an-post-in-php, Last visited 03.07.2017.

[22] stackoverflow, https://stackoverflow.com/questions/15461786/pass-javascript-variable-to-php-via-ajax, Last visited 03.07.2017.

[23] stackoverflow, https://stackoverflow.com/questions/14976495/get-selected-option-text-with-javascript, Last visited 03.07.2017.

[24] stackoverflow, https://stackoverflow.com/questions/11821261/how-to-get-all-selected-values-from-select-multiple-multiple, Last visited 03.07.2017.

[25] stackoverflow, https://stackoverflow.com/questions/10960471/passing-data-back-from-php-to-ajax, Last visited 03.07.2017.

[26] stackoverflow, https://stackoverflow.com/questions/1571648/html-tab-space-instead-of-multiple-non-breaking-spaces-nbsp, Last visited 03.07.2017.

[27] stackoverflow, https://stackoverflow.com/questions/30269461/uncaught-typeerror-cannot-use-in-operator-to-search-for-length-in, Last visited 03.07.2017.

[28] W3schools, https://www.w3schools.com/html/html_tables.asp, Last visited 03.07.2017.

[29] Sitepoint, https://www.sitepoint.com/community/t/how-to-properly-populate-html-table-with-jquery-ajax-using-json-formatted-data/202187/5, Last visited 03.07.2017.

[30] stackoverflow, https://stackoverflow.com/questions/17724017/using-jquery-to-build-table-rows-from-ajax-responsejson, Last visited 03.07.2017.

[31] stackoverflow, https://stackoverflow.com/questions/10960471/passing-data-back-from-php-to-ajax, Last visited 03.07.2017.

[32] stackoverflow, https://stackoverflow.com/questions/9816889/how-to-echo-an-array-in-php, Last visited 03.07.2017.

[33] stackoverflow, https://stackoverflow.com/questions/25295489/holding-the-table-name-in-a-variable-inside-a-stored-procedure-using-a-prepared Last visited 03.07.2017.

[34] stackoverflow, https://stackoverflow.com/questions/15461786/pass-javascript-variable-to-php-via-ajax, Last visited 03.07.2017.

[35] PHP, http://php.net/manual, Last visited 03.07.2017.

[36] stackoverflow, https://stackoverflow.com/questions/8336858/how-to-combine-two-strings-together-in-php, Last visited 03.07.2017.

[37] stackoverflow, https://stackoverflow.com/questions/6239766/compare-variables-php, Last visited 03.07.2017.

[38] stackoverflow, https://stackoverflow.com/questions/4915753/how-can-i-remove-3-characters-at-the-end-of-a-string-in-php, Last visited 03.07.2017.

[39] stackoverflow, https://stackoverflow.com/questions/20387095/uncaught-typeerror-cannot-read-property-length-of-null-in-jquery, Last visited 03.07.2017.

[40] stackoverflow, https://stackoverflow.com/questions/18798353/how-can-i-pass-input-parameter-with-single-quote-in-stored-procedure, Last visited 03.07.2017.

[41] stackoverflow, https://stackoverflow.com/questions/2440506/how-to-check-if-an-array-value-exists, Last visited 03.07.2017.

[42] stackoverflow, https://stackoverflow.com/questions/1100354/how-can-i-echo-html-in-php, Last visited 03.07.2017.

[43] stackoverflow, https://stackoverflow.com/questions/8336858/how-to-combine-two-strings-together-in-php, Last visited 03.07.2017.

[44] W3schools, https://www.w3schools.com/php/php_looping_for.asp, Last visited 03.07.2017.

[45] stackoverflow, https://stackoverflow.com/questions/4414623/loop-through-an-array-php, Last visited 03.07.2017.

[46] stackoverflow, https://stackoverflow.com/questions/11441369/php-string-concatenation, Last visited 03.07.2017.

[47] Tutorialspoint, https://www.tutorialspoint.com/php/php_strings.htm, Last visited 03.07.2017.

[48] stackoverflow, https://stackoverflow.com/questions/370013/jquery-delete-all-table-rows-except-first, Last visited 03.07.2017.

[49] W3schools, https://www.w3schools.com/html/html_tables.asp, Last visited 03.07.2017.

[50] stackoverflow, https://stackoverflow.com/questions/7271490/delete-all-rows-in-an-html-table, Last visited 03.07.2017.

[51] Stateexchange, https://tex.stackexchange.com/questions/8652/what-does-t-and-ht-mean, Last visited 03.07.2017.

[52] namsu, https://www.namsu.de/latex/kapitel6.html, Last visited 03.07.2017.

[53] Balsamiq, https://docs.balsamiq.com/desktop/datagrids/, Last visited 03.07.2017.

[54] stackoverflow, https://stackoverflow.com/questions/6896592/is-it-possible-to-save-html-page-as-pdf-using-javascript-or-jquery, Last visited 03.07.2017.

[55] W3schools, https://www.w3schools.com/css/css_margin.asp, Last visited 03.07.2017.

[56] stackoverflow, https://stackoverflow.com/questions/22793061/making-a-select-show-as-a-scrollable-list-with-only-one-selection, Last visited 03.07.2017.

[57] Mediaeventt, https://www.mediaevent.de/css/margin.html, Last visited 03.07.2017.

[58] HTML3Canvas, https://github.com/niklasvh/html2canvas, Last visited 03.07.2017.

[59] stackoverflow, https://stackoverflow.com/questions/28193487/how-to-use-addhtml-function-in-jspdf, Last visited 03.07.2017.

[60] stackoverflow, https://stackoverflow.com/questions/30319763/how-to-set-font-size-of-exported-table-in-jspdf-js, Last visited 03.07.2017.

[61] C-Sharpcorner, http://www.c-sharpcorner.com/UploadFile/b629e0/creating-pdf-from-html-through-jspdf/, Last visited 03.07.2017.

[62] stackoverflow, https://stackoverflow.com/questions/23104008/where-to-change-default-pdf-page-width-and-font-size-in-jspdf-debug-js, Last visited 03.07.2017.

[63] stackoverflow, https://stackoverflow.com/questions/10211145/getting-current-date-and-time-in-javascript, Last visited 03.07.2017.

[64] stackoverflow, https://stackoverflow.com/questions/10211145/getting-current-date-and-time-in-javascript, Last visited 03.07.2017.

[65] stackoverflow, https://stackoverflow.com/questions/12934144/how-to-reload-refresh-jquery-datatable, Last visited 03.07.2017.

[66] stackoverflow, https://stackoverflow.com/questions/32852388/jquery-datatables-no-data-available-in-table, Last visited 03.07.2017.

[67] DataTabels, https://datatables.net, Last visited 03.07.2017.

[68] stackoverflow, https://stackoverflow.com/questions/14267781/sorting-html-table-with-javascript, Last visited 03.07.2017.

[69] stackoverflow, https://stackoverflow.com/questions/27047365/jspdf-addhtml-method-not-working, Last visited 03.07.2017.

[70] stackoverflow, https://stackoverflow.com/questions/16858954/how-to-properly-use-jspdf-library, Last visited 03.07.2017.

[71] stackoverflow, https://stackoverflow.com/questions/27545394/how-to-save-jspdf-file-into-specific-folder, Last visited 03.07.2017.

[72] stackoverflow, https://stackoverflow.com/questions/10042413/how-to-place-the-table-at-the-middle-of-the-webpage, Last visited 03.07.2017.

[73] stackoverflow, https://stackoverflow.com/questions/8986260/how-to-align-center-the-text-in-html-table-row, Last visited 03.07.2017.

[74] W3schools, https://www.w3schools.com/tags/att_table_border.asp, Last visited 03.07.2017.

[75] stackoverflow, https://stackoverflow.com/questions/24403732/check-if-array-is-empty-does-not-exist-js, Last visited 03.07.2017.

[76] Reddit, https://www.reddit.com/r/javascript/comments/3fieit/what_is_the_best_method_to_check_if_a_variable_is/, Last visited 03.07.2017.

[77] stackoverflow, https://stackoverflow.com/questions/6003884/how-do-i-check-for-null-values-in-javascript, Last visited 03.07.2017.

[78] PHP, http://php.net/manual/en/function.is-null.php, Last visited 03.07.2017.

[79] stackoverflow, https://stackoverflow.com/questions/1992114/how-do-you-create-a-hidden-div-that-doesnt-create-a-line-break-or-horizontal-sp, Last visited 03.07.2017.

[80] stackoverflow, https://stackoverflow.com/questions/3239202/whats-the-correct-way-to-hide-the-h1-tag-and-not-be-banned-from-google, Last visited 03.07.2017.

[81] W3schools, https://www.w3schools.com/jsref/dom_obj_text.asp, Last visited 03.07.2017.

[82] stackoverflow, https://stackoverflow.com/questions/23851337/check-if-ajax-response-data-is-empty-blank-null-undefined-0, Last visited 03.07.2017.

[83] Coderanch, https://coderanch.com/t/122806/hide-unhide-table-html, Last visited 03.07.2017.

[84] stackoverflow, https://stackoverflow.com/questions/6910349/hiding-table-data-using-div-style-displaynone, Last visited 03.07.2017.

[85] Tutorialspoint, https://www.tutorialspoint.com/php/php_strings.htm, Last visited 03.07.2017.

[86] stackoverflow, https://stackoverflow.com/questions/3533474/php-foreach-put-each-of-the-loop-result-in-one-variable, Last visited 03.07.2017.

[87] stackoverflow, https://stackoverflow.com/questions/9848346/how-to-escape-quotes-apostrophe-in-php, Last visited 03.07.2017.

Abbreviations

CoMaDa	Configuration, Management and Data Handling Framework
CSG	Communication System Group
GUI	Graphical User Interface
HTML	Hypertext Markup Language
PHP	PHP: Hypertext Preprocessor
SecureWSN	Secure Wireless Sensor Network
SQL	Structured Query Language
UZH	University of Zurich
WebMaDa	Web-based Mobile Access and Data Handling Framework
WSN	Wireless Sensor Network

List of Figures

List of Tables

Appendix A

Contents of the CD

The attached CD contains the following files and directories:

- thesis.pdf: PDF of the thesis
- code: Files that are used on WebMaDa
- presentation: Final presentation