Design Guide SESAME-S Tablet

Author: Ulrich Lehner
2011-12-20

1 Introduction

This document is a style guide for the energy feedback monitor as part of the SESAME-S project. The overall goal for this app is to provide an energy feedback monitor and the integration for the Power Management Service. Since the test trials will compare different configurations of the app (e.g. detailed vs. abstract, small screen vs. large screen, etc.) the final presentations should be a composition of different widgets (further stated as *fragments*) defined by a *layout* (e.g. more detailed view, displayed on a Smartphone, etc.).

1.1 Units

The provided unit for energy consumption from the cloud service provided by E-Smart is (as known so far) Watt. Since for an average person this value may be meaningless an alternative conversion into a currency should be given (Euro). This might not be that trivial, since converting depends e.g. on the provider or the time (day vs. night), but this information should be available in some form (maybe E-Smart?). However the app should be prepared to show following units:

1. **kW** as the currently needed energy
2. **kWh** as the consumption so far
3. **Euro** as money spent so far or (for real time values) and as the electric price for the current consumption (again this must be clarified)

2 Fragments

The app will mainly be a combination of different fragments displaying data about energy consumption or allowing users to interact with the device or services. Most of these fragments can be visualized with the library *AChartEngine*[1] for the Android platform. Some of these fragments may be displayed with more options (referred as *extended features*, e.g. displaying a historical values).

2.1 Common features

All fragments should be designed in a way that they can consist in an own frame (e.g. to style it with a border, background color etc.). Analog to a UI window a fragment should have a title displayed at the top.

[1] http://www.achartengine.org/

2.2 Real time energy consumption

This fragment should visualize real time consumption in form of a dial chart (like the metaphor of an analog tachometer, as shown in Figure 1 Tachometer). It is intended to give a real time feedback of the current energy consumption without distracting from other values. In the case of BBS Kirchdorf real time means '15 minute intervals', so changes from one value to another will be very discrete (this may look like the hand will jump from value to another). However to avoid this situation, a value change should be animated (since this user control will be customized anyway, responsibility of this behavior will be at FTW).

Figure 1 Tachometer

2.2.1 Input parameters

The latest measured LoadProfile (*DataValue*) from a cloud service for a specific room (e.g. *EDV1*) – since the parameters require a time range this must be tricked somehow.

2.2.2 Display parameters

1. Color for a handle (simplified as constants like *RED* or *BLUE*)
2. To determine the range following attempts are possible (since this fragment should be displayed for each room the ranges should be the same for all)
    a. *Static* (the value of 10% is just an assumption, there may be better values)
        i. Lower bound: min value of overall consumption of all rooms – 10% rounded to a 'pretty' value
        ii. Upper bound: same as above from max side + 10%
    b. *Dynamic*
    Same rules as above but done programmatically
3. Unit of displayed data
4. Label for the room name

2.2.3 Output

The latest received *DataValue* will be displayed as value (handle position) with a *DialChart*. For further visual enhancement this chart will be extended to display the chart in a more elegant way, but the functionality remains the same.

2.2.4 Extended features

For more detailed information a second value should be displayed (in a less dominant color/style then the main handle) to compare the consumption. Following values are possible and should be adjustable:

1. Overall average consumption of a room from the beginning of a given date (in case of BBS Kirchdorf the begin of the trial)
2. Consumption of the previous (labor) day at the same time (e.g. now=2011-12-20 17:00 then consumption of 2011-12-19 17:00)
3. Consumption of the same day at the previous week at the same time (e.g. now=2011-12-20 17:00 then consumption of 2011-12-13 17:00)

2.2.5 Mockup

See Figure 1 Tachometer.

2.3 Real time overall consumption

To provide information of how much energy was consumed from a given date (in this case the start of a test trial), a 'power meter'-like visualization should be implemented. This simply gives a nice overview and should later be styled to look more compelling (comparing Figure 2 Digits from a power meter this gives a realistic look of the digits). For the first iteration of implementation this should be done in simplified way, e.g. in a *TextBox* (which will later be styled, derived etc.). A change of a digit should be animated (responsibility of FTW).

Figure 2 Digits from a power meter

2.3.1 Input parameters

This fragment should receive its data from the LoadProfile (maybe DailyConsumption will be the better choice) from a given date (should be flexible, but probably the start date of a trial) to the current time and sum up the *DataValues*.

2.3.2 Display parameters

1. Unit of displayed data.

2.3.3 Output

Current consumption

2.3.4 Mockup

See Figure 2 Digits from a power meter.

2.4 Real time line chart

This fragment should visualize the real time consumption on a daily basis displayed as a line chart for a specific room.

2.4.1 Input parameters

Data should be easily accessed via the *LoadProfile* function for a specific room for the current day.

2.4.2 Display parameters

1. Time range (basically it makes sense to set these bounds a little bit lower/higher then a regular school day, e.g. from 7am to 8pm)
2. Room (Measure Point)
3. Unit of displayed data

2.4.3 Output

This fragment shows the real time consumption (see Figure 3 Simple line chart) and optionally previous consumption (see Figure 4 Comparing line chart and Figure 5 Combined line chart) for a better understanding.

2.4.4 Extended Features

It is also interesting to compare the current situation with past consumptions. This can be implemented in the following ways:

1. On a per-day basis
    a. *Comparing one day:* consumption of previous (labor) day. This can visualized by drawing a second graph behind the actual in a less dominant way (e.g. make it less opaque). See Figure 4 Comparing line chart.
    b. *Comparing more days:* Consumption of previous (labor) days. Can be achieved the same way as above (layering graphs and decrease opaque – this of course is limited soon). See Figure 5 Combined line chart.
2. On a per-week basis
    a. *Comparing previous week:* consumption of previous week, displayed as above
    b. *Comparing more weeks:* consumption of previous weeks, displayed as above

2.4.5 Interaction

In case of this fragment an interaction from the user is required to display the chart of a specific room. This should be accessible by a *TabView* or *Buttons* to select a room. This interaction may motivate the user to actively engage with the app (see Figure 3 Simple line chart and Figure 4 Comparing line chart).

Also to give more possibilities to compare different values (in case of the extended feature set 1a and 2a) a *DropDown* list should allow the user to set the comparing value (in case of 1a a list of previous daya respectively a list of previous weeks in 2a) – see Figure 4 Comparing line chart. Cases 1b and 2b are intentionally combined solutions and shouldn't offer the user to choose.

2.4.6 Mockup

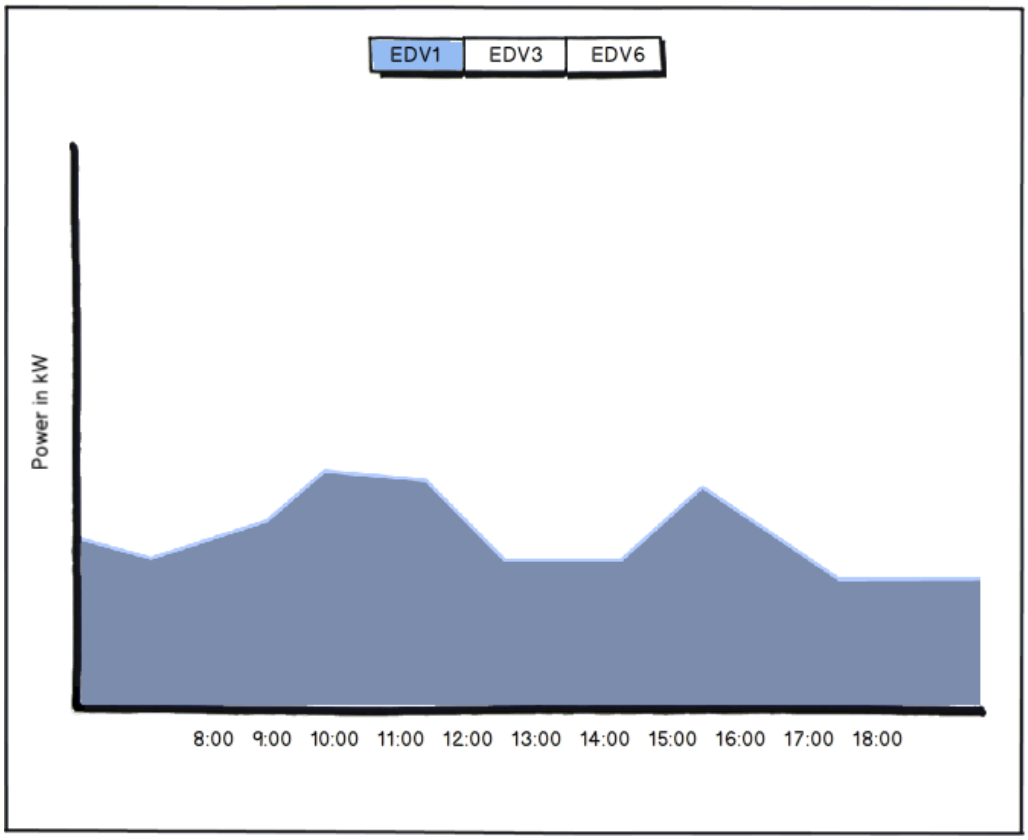

Figure 3 Simple line chart

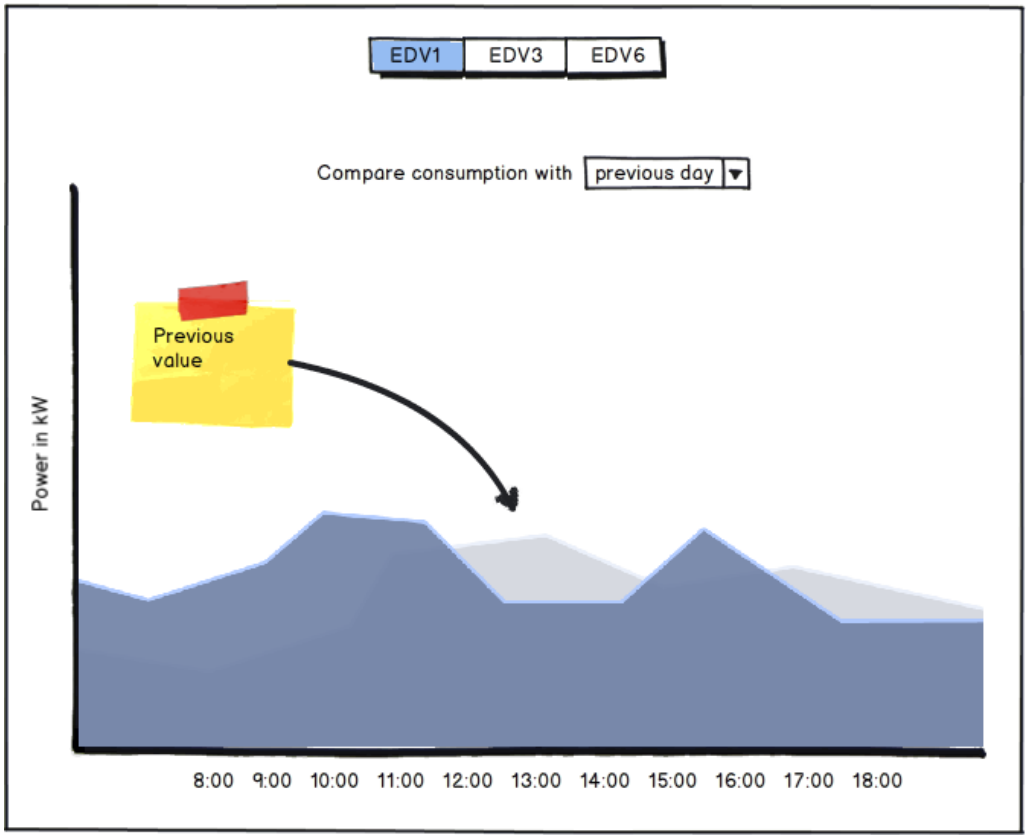

Figure 4 Comparing line chart

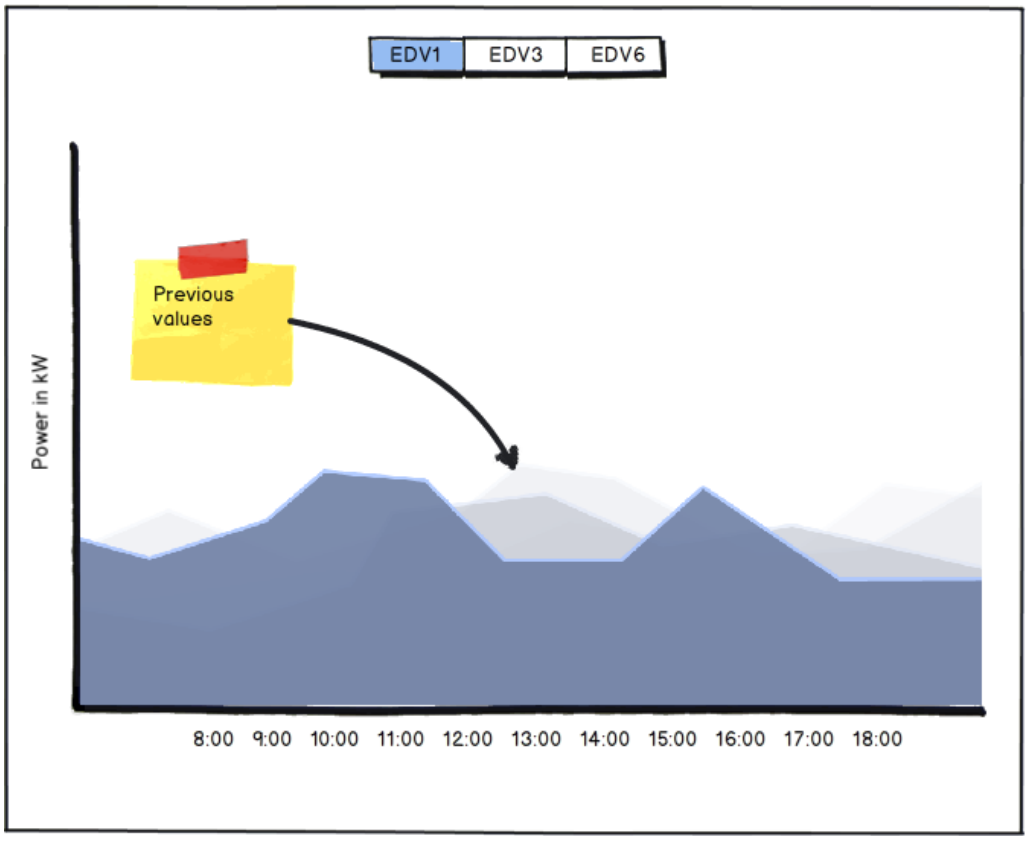

Figure 5 Combined line chart

2.5 Overall real time line chart

This is basically the same visualization as 2.4 Real time line chart with the difference that there is no distinction between the rooms (Measure Points). Instead one line graph should be displayed with the average consumptions of all rooms. With this simplification, the requirements for this fragment are a the same as in 2.4 Real time line chart and therefore it may be efficient to combine these features (since the comparison methods a actually the same).

2.5.1 Input parameters

Average *LoadProfile* of all *MeasurePoints* for the current day.

2.5.2 Display parameters

See above (except MeasurePoint)

2.5.3 Output

See above.

2.5.4 Extended Features

Since the consumption is composed of each room, an option to see the itemized consumption would be a nice feature to attract attention. See Figure 6 Compound line graph.

2.5.5 Interaction

By pressing onto the line a popup containing the information of the specific x value should be displayed (e.g. as bar chart or pie chart). See Figure 6 Compound line graph.

2.5.6 Mockup

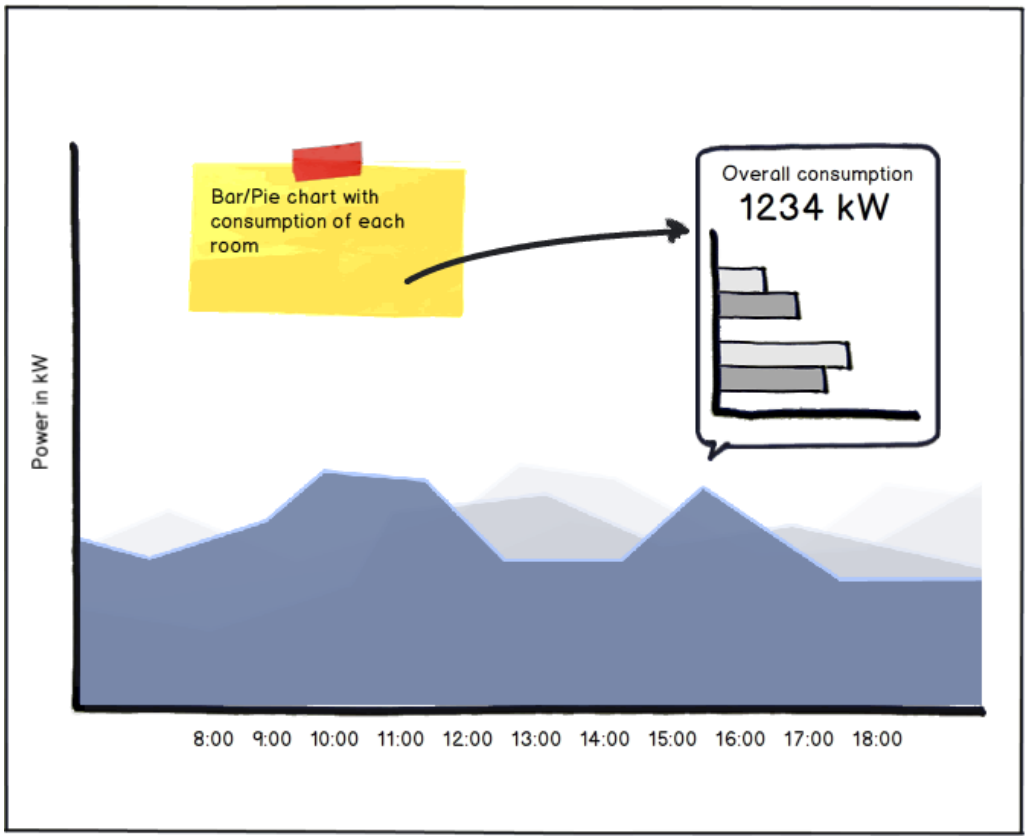

Figure 6 Compound line graph

2.6 Success indicator

This fragment is supposed to be combined with a line chart, showing how the current behavior is performing (e.g. current consumption is beneath the average consumption will result in a positive indicator vice-versa). It should be able to display three states: positive, neutral and negative (see Figure 7 Success indicator).

2.6.1 Input parameters

The input should correspond to 2.4 Real time line chart respectively 0 It is also interesting to compare the current situation with past consumptions. This can be implemented in the following ways:

3. On a per-day basis
    a. *Comparing one day:* consumption of previous (labor) day. This can visualized by drawing a second graph behind the actual in a less dominant way (e.g. make it less opaque). See Figure 4 Comparing line chart.

   b. *Comparing more days:* Consumption of previous (labor) days. Can be achieved the same way as above (layering graphs and decrease opaque – this of course is limited soon). See Figure 5 Combined line chart.
4. On a per-week basis
   a. *Comparing previous week:* consumption of previous week, displayed as above
   b. *Comparing more weeks:* consumption of previous weeks, displayed as above

2.6.2 Interaction

In case of this fragment an interaction from the user is required to display the chart of a specific room. This should be accessible by a *TabView* or *Buttons* to select a room. This interaction may motivate the user to actively engage with the app (see Figure 3 Simple line chart and Figure 4 Comparing line chart).

Also to give more possibilities to compare different values (in case of the extended feature set 1a and 2a) a *DropDown* list should allow the user to set the comparing value (in case of 1a a list of previous daya respectively a list of previous weeks in 2a) – see Figure 4 Comparing line chart. Cases 1b and 2b are intentionally combined solutions and shouldn't offer the user to choose.

2.6.3 Mockup

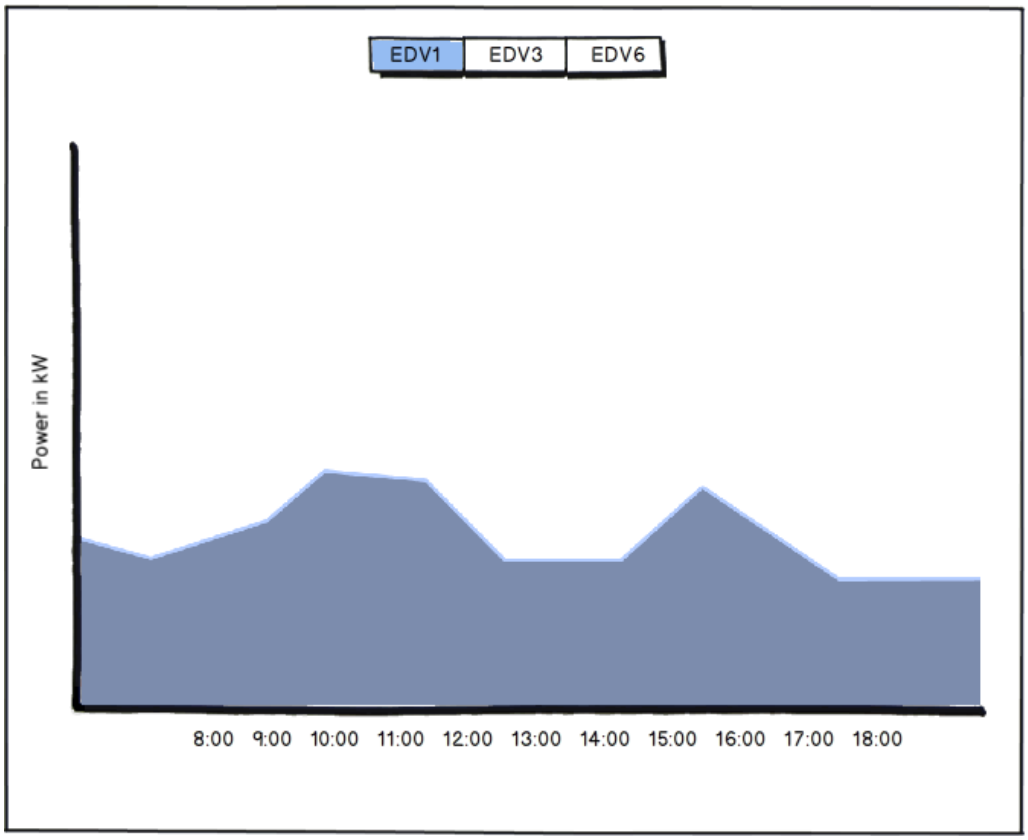

Figure 3 Simple line chart

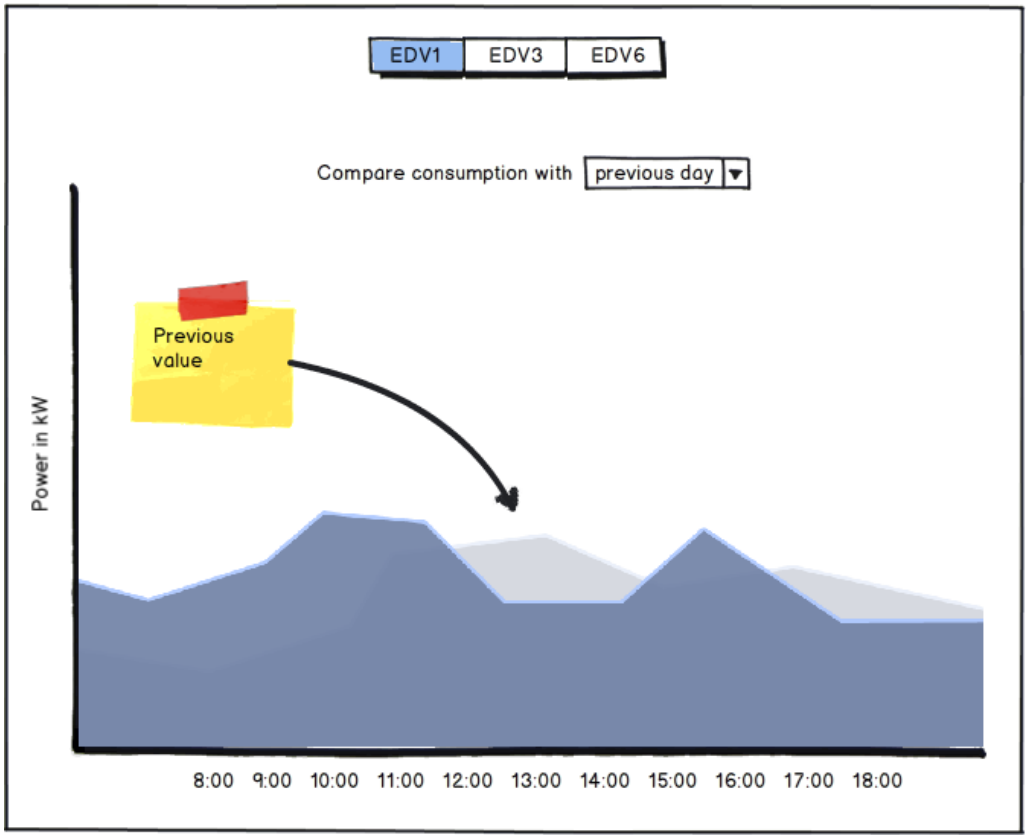

Figure 4 Comparing line chart

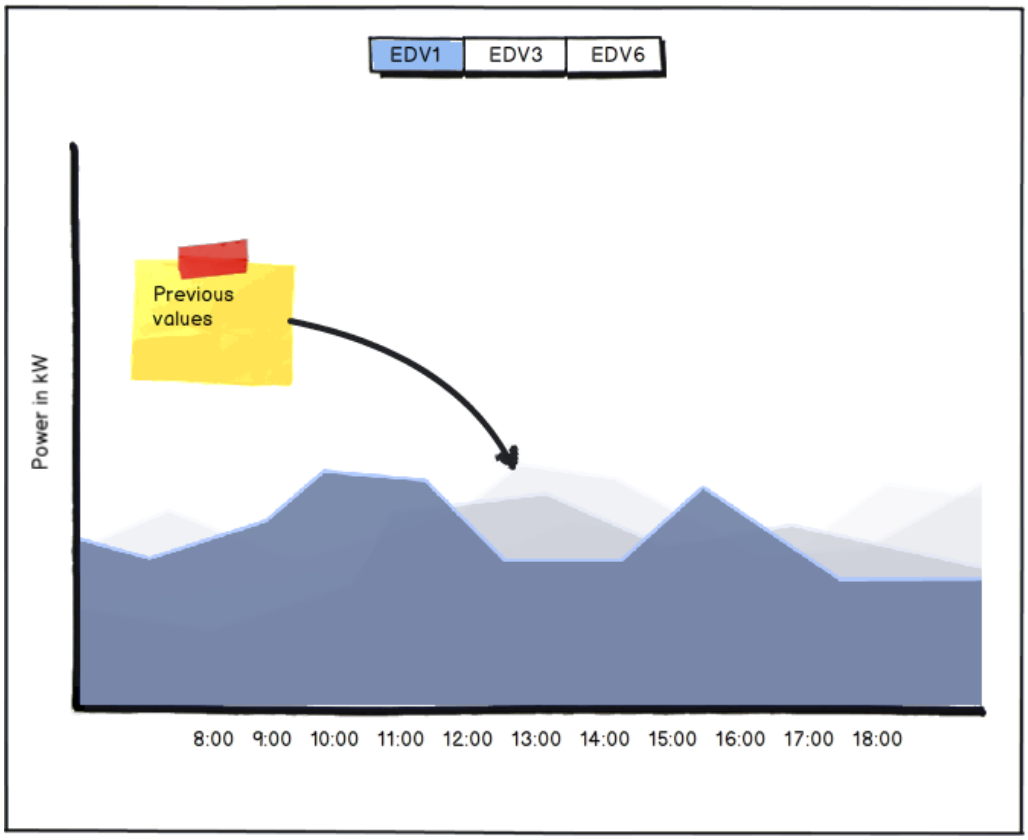

Figure 5 Combined line chart

Overall real time line chart since this indicates the behavior of this graph. Also the display of this fragment only makes sense if there's a value to compare.

2.6.4 Display parameters

1. Display type (e.g. arrow, smiley – but can be static at first)

2.6.5 Mockup

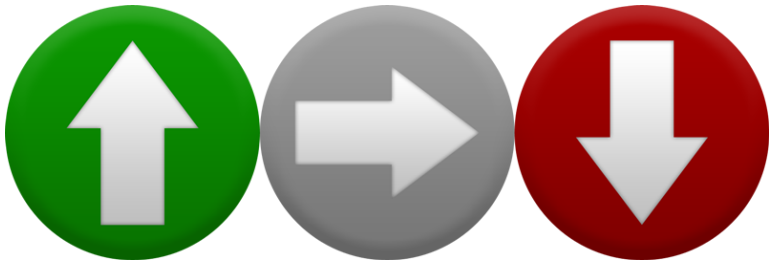

Figure 7 Success indicator

2.7 Power management service

The existing functionality of the Power Management Service should be implemented into this app. Since this feature is well described in another document (‘*Power Management Service App*’) there won’t be any further explanations of the features.

Still unclear, but PCs in the list should be associated with the corresponding room.

However there may be the option of a ‘light’ version (e.g. reduced possibility for remote shut-down), which is not yet defined.

2.8 Ambient signal

The purpose for this fragment is to attract attention when something is happening (or there’s a relevant notification). It is meant to work as a neutral background color but has the ability to pulsate (slow, smooth and eye-friendly transitions) in a specified color. Since a layout option could be semitransparent, this would fit very well (see Figure 8 Transparent layout: the background can remain still as nothing happens and begin to pulsate if there’s something happening).

Figure 8 Transparent layout

2.8.1 Display parameters

1. A Boolean value indicating that the fragment should pulsate (or remain still otherwise)
2. Fixed base color value.

3 Layouts

For the planned design comparison study in February/March 2012 the energy monitor will be shown with a different granularity of visible information and possibility for interaction to the pupils. They mainly differ in their information density and interaction possibilities to meet the needs of various school contexts (e.g. operating during lecture, display mounted in hallway). These layouts are a combination of the above-described *Fragments*.

Note: graphic design of layouts may change.

3.1 Low information density

This design is an abstract visualization of energy feedback and is based on *ambient lights*. It is indented not to raise any attention during normal operation and therefore suits the need of a display in a school during a lecture (pupils should not be distracted). Also the possibilities of interaction are very restricted. In case of a notification (in HAK-context e.g. computer is running unattended) the display may attract attention. The implementation of this visualization should be achieved with the *Ambient Signal* fragment.

3.1.1 Calm operation mode

During normal operation (no notifications etc.) the display should only show dimmed color (fading color changes). Shades of blue are believed to be relaxing. The following example can be found on kuler.adobe.com (theme *Baby Ocean*):

#415469	#607D91	#9FC1CD	#BFE3E5	#E7F5F8

Table 1 Shades of blue

These colors should fade in a smooth transition (time between two colors might be 5 seconds) and the given or random order. Beside the monitor brightness of the tabled may also be dimmed down.

3.1.2 Notification mode

Contrary to the *Calm operation mode* a notification should attract attention. The display should act more 'intrusive' using signal colors and pulsating with a higher frequency. This should last until someone is checking the notification. Also the time from receiving the notification until checking it should be considered: that means frequency and display brightness should increase over time.

#FFF37E	#EDDB53	#E3CA23	#A68D22	#9C840A

Table 2 Warning colors

Unlocking the display
It is required for the user to interact with the display to receive the notification. Two ways of implementations seem possible:

1. Using **Face recognition**: this option would require the user to stand in front of the display. The tabled recognizes that a person is starring at the display and therefore can view the notification.
2. Using **User input**: the user needs to push a button on to display to view the notification (see Figure 9 Warning with button)

Figure 9 Warning with button

The received message should be kept simple and invite to user to take action (e.g. look in the specific room and check what's happening). Also the color animation should be stopped when viewing the notification. Figure 10 Notification shows a concept of the notification, containing a *TextView* and a *Button*.

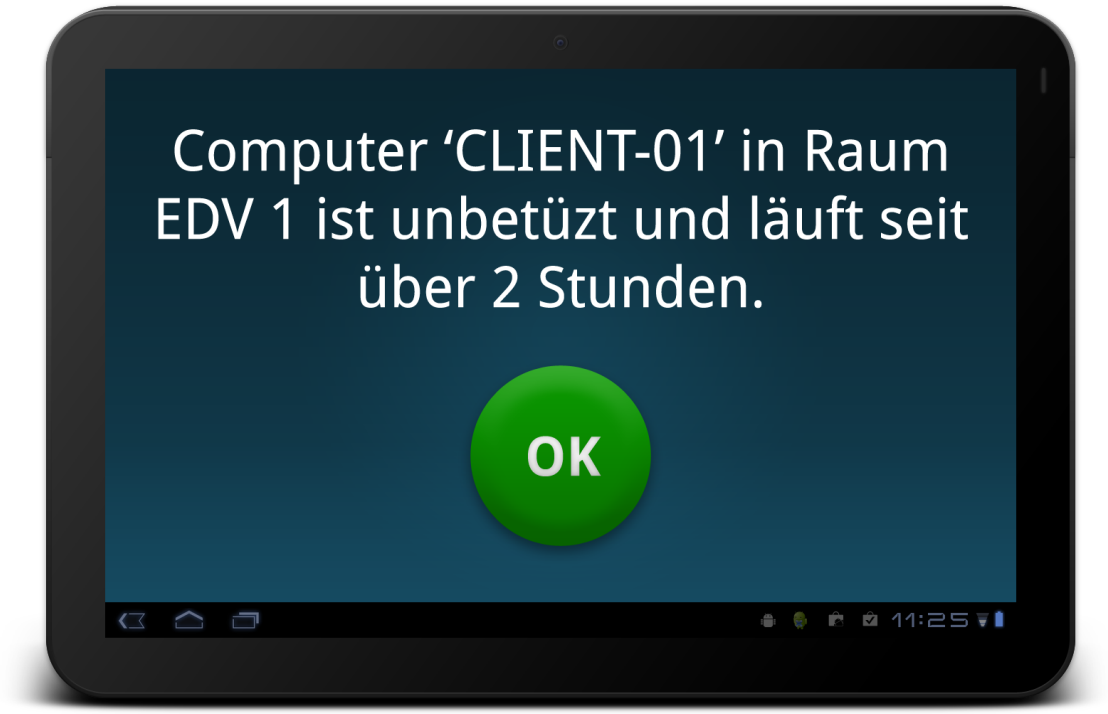

Figure 10 Notification

By pushing the *OK Button* the display will go back into *Calm operation mode* and the notification can be marked as *read* internally.

3.2 Medium information density

Like the *Low information density* visualization, this layout is intended to operate during a lesson but will show more accurate feedback. The idea of this layout is based on *public screens* where information must be conveyed quickly (because of the smaller attention span). The overall idea is to show single information at one time (e.g. a graph of real time energy consumption as line chart using the whole screen). The main challenges are to convey the information in a simple way and to make and readable from larger viewing distances.

Example: assuming a classroom with a size of 6 x 6 m (= 36 m^2) means the largest viewing angle might be approx. 9m. Dealing with a screen size of 10,1″ (like the Acer Iconia Tab) means that the font-size should be at least 4,5 cm on the screen to be readable from most spots of the classroom (a font size should be 1/200 of the viewing distance).

Also the granularity of the information doesn’t have to be very detailed (e.g. using a compounded consumption for all computer rooms). Following section describes the needed visualization and the used data. Each representation should fill the whole screen and stay for some

time (e.g. 30 seconds). To make the transition more appealing an animation should be applied between two visualizations (e.g. blending two screens or translating the screen horizontally – compare Figure 11 Blending visualizations and Figure 12 Translating visualizations).

Figure 11 Blending visualizations

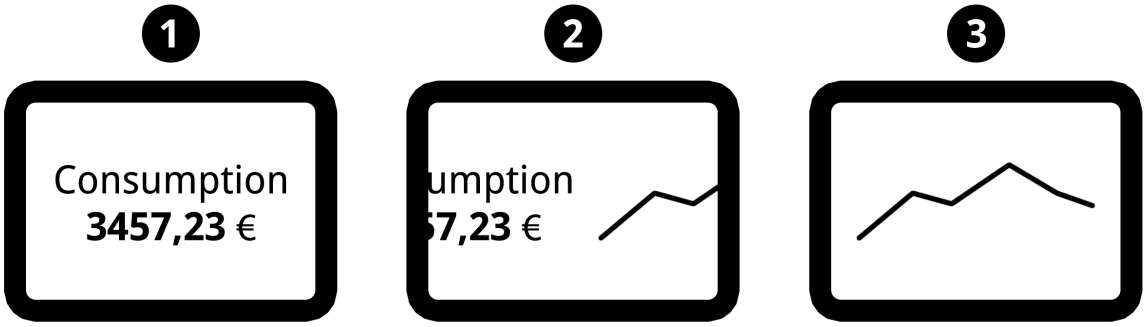

Figure 12 Translating visualizations

Following visualizations should be displayed:

1. Real time consumption as a **line chart** for each room combined with a **success indicator**.
2. Real time and overall consumption as a **dial chart** in combination with a **numeric** view, that indicates the overall consumption since a given date (e.g. since the start of a month)

At the moment there are three measurement places (EDV 1, EDV 3 and EDV 6), each having two ways to visualize the energy consumption.

3.2.1 Real time consumption (line chart)

As Figure 13 shows, the layout for this visualization is very simple: the caption should stay on the top (with a large font size to be readable) and the main focus should be on a hight-contrast line-chart. Bottom will be used for the labels (also large font-size) and the success indicator should stick on the top right, if available.

Figure 13 Highdef mockup for line chart

3.2.2 Real time consumption (dial chart)

This visualization should focus on the current consumption of a room including the overall consumption on a given date, based on the metaphor of an energy meter. This fragment should be labeled with the room on the top and a label indicating the start date of the measurements on the bottom. The dial chart will display the current power consumption, while the numeric value will display the accumulated consumption of a given date for a room. Figure 14 shows the visual representation.

Figure 14 Highdef mockup of dial chart

3.3 High information density

Combining all available information into a finer granularity and giving more possibilities to interact with the display results into the *High information density* visualization. The overall goal for this layout is to invite the user to deal with the display. It should provide real time data as well as overall consumption and making it possible to compare historical consumption. In HAK Kirchdorf a *Power Management Service* is installed which should also be accessible on the tablet. The layout of this representation will be discussed in this section.

The screen is divided into following sections (see Figure 15 Mockup for high information density layout):

1. Scalar real time energy consumption
2. Overall energy consumption
3. Real time energy consumption as line graph
4. Power Management

Fragments (1) and (2) offer mainly a quick overview for a specific room (since there a are only three at the moment this fits on the screen – more rooms may require a different approach).

Figure 15 Mockup for high information density layout

Figure 16 Highdef mockup for high information density layout

3.3.1 Scalar real time consumption

This fragments has the same functionality as in the medium information density layout (see Real time consumption (dial chart)) and is therefore a minified version. There may be one fragment for each room (EDV 1, EDV 3 and EDV 6).

Optional features

1. Since EZAN sensors also provide room temperature and humidity, this data can be visualized on the bottom of this fragment.
2. A second pointer may indicate another consumption, e.g. an average consumption or an historical comparison value.

3.3.2 Overall energy consumption

This fragment should be combined with the previous dial chart visualization and follow the functionality described Real time consumption (dial chart) – an accumulated power consumption.

3.3.3 Real time energy consumption as line graph

To give an overview of the real time power consumption, a line graph will be used. Following use cases should be handled:

1. Comparison of consumption for each room
2. Comparison of historical consumption for a specific room

To choose either of these options, the user should pick one via a tab bar at the top of this fragment.

Comparison of consumption for each room
A checkbox list should be used to select a room. The line charts of these rooms should be overlayed (as shown in Figure 16).

Comparison of historical consumption for a specific room
To compare the historical consumption, following aspects are important to implement:

1. Compare consumption with same day of previous week(s) – checkbox list of previous weeks
2. Compare consumption with other days of a week – checkbox list of previous days
3. Give an overview of the consumption within a week – radio button list of calendar week

TODO: UI for dealing with graph.

3.3.4 Power Management

The Power Management mainly follows the functionality of the previous built Smartphone app. Additionally list items should be rooms (instead of a list of all computers). To indicate a new notification of this service, a label with the number of notifications should appear (see Figure 16).

To navigate to all computers in a specific room a **separate screen** (or overlay) with a list of computers should appear (see Figure 17). This screen is very similar to the Android phone client for the Power Management Service. Instead of presenting a long list, the screen size of the tablet allows to view to separate lists divided into two columns. To navigate to this screen, it is necessary to touch one of the list items in the room list.

TODO: buttons for power on/off computer.

Figure 17 Power Management computer list