I. Installation

1. Connecting the Ekey Scanners

See the list of the supported ekey FSX scanners below:

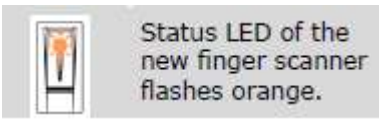
101234	ekey FSX OM E REL
101235	ekey FSX OM E RFID
101236	ekey FSX OM E RFID REL
101237	ekey FSX OM E
101384	ekey FSX IN
101388	ekey FSX IN RFID
101427	ekey FSX WM
101428	ekey FSX WM RFID
101429	ekey FSX WM REL
101430	ekey FSX WM RFID REL
101444	ekey FSX OM I
101767	ekey FSX OM I RFID
101813	ekey FSX OM I REL
101814	ekey FSX OM I RFID REL
102023	ekey FSX OM E BL RFID
102024	ekey FSX OM E BL
102025	ekey FSX OM I BL
102026	ekey FSX OM I BL RFID

- Note that the "ekey home" and "ekey net" scanners are NOT supported!

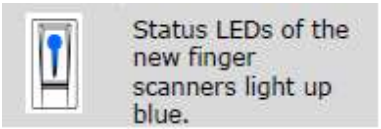
Connect the ekey FSX scanner to the one of the available RS-485 ports.

RS485 B <--> ekey scanner pin1 (green)
RS485 A <--> ekey scanner pin2 (yellow)
- 12-24V DC <--> ekey scanner pin3 (brown)
+12-24V DC <--> ekey scanner pin4 (white)

A group of up to 8 ekey FSX scanners can be connected to a single RS-485 line.
Scanners should be automatically detected by the application – see the *tab "Scanners"*.
If the connection to the RS485 line is successful, the scanner's central LED light changes from a blinking orange



to a steady blue colour:



The instruction manual is available on https://openrb.com/ekey-biometric-scanner-interconnection-with-lm/

- For LM devices with old CPU (LM2, LM3, LM4) is not recommended to use the built-in rs485 ports due to serial communications instability.
- The RS485 line can't be used together with the Modbus devices.

If you have no free LM RS485 native ports there is possible to use external "USB to RS485" converters with a FTDI chips FT232RL, FT232R, FT232, but is not compatible with other serial chips like CH340

The following USB - RS485 converters are succesfully tested:
1) FTDI FT232 chip: the original ekey USB Converter:
https://www.ekey.net/ajax/datasheet.php?lg=en&id=100433

2) FTDI FT232R chip: usb-rs485-we:
https://www.ftdichip.com/Support/Documents/DataSheets/Cables/DS_USB_RS485_CABLES.pdf

3) FTDI FT232RL chip: USB zu TTL RS485 Serial Konverter Adapter:
https://www.ebay.de/itm/123649346462

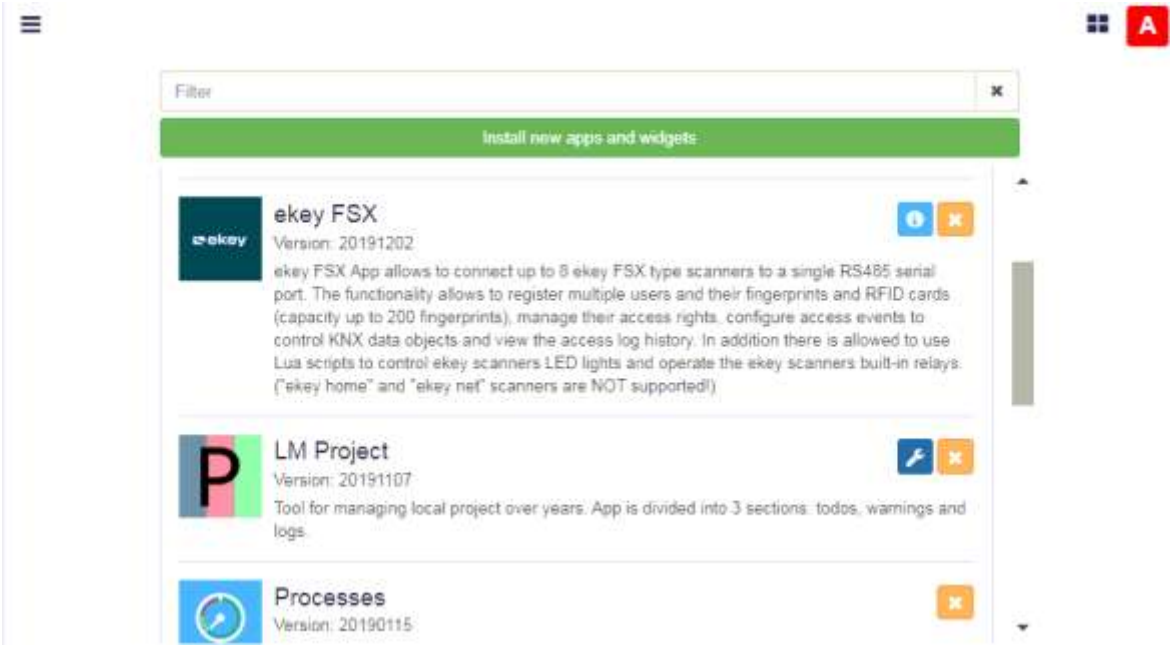
2. Installing the "ekey FSX app"

2.1. Download and install the last version installation package from:

http://www.ekey.lv/logicmachine/ekey-app/ekey-app-20200711.ipk

or

2.2. Install it from the LogicMachine App store:



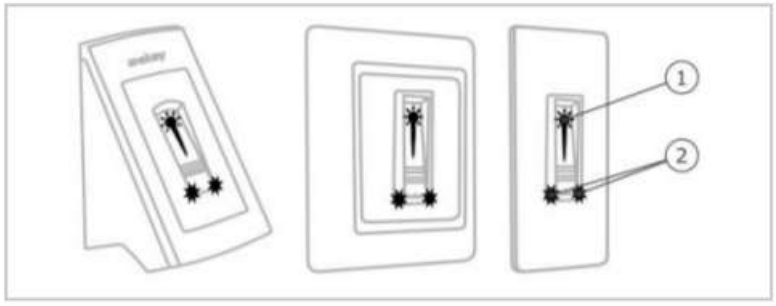
3. Installing the "ekey TA app"

2.3. You can download the separate "ekey TA app" to expand this app with a "Time attendance" functionality
http://www.ekey.lv/logicmachine/ekey-app/ekey-TA-20200711.ipk

4. Optical signals on the finger scanner

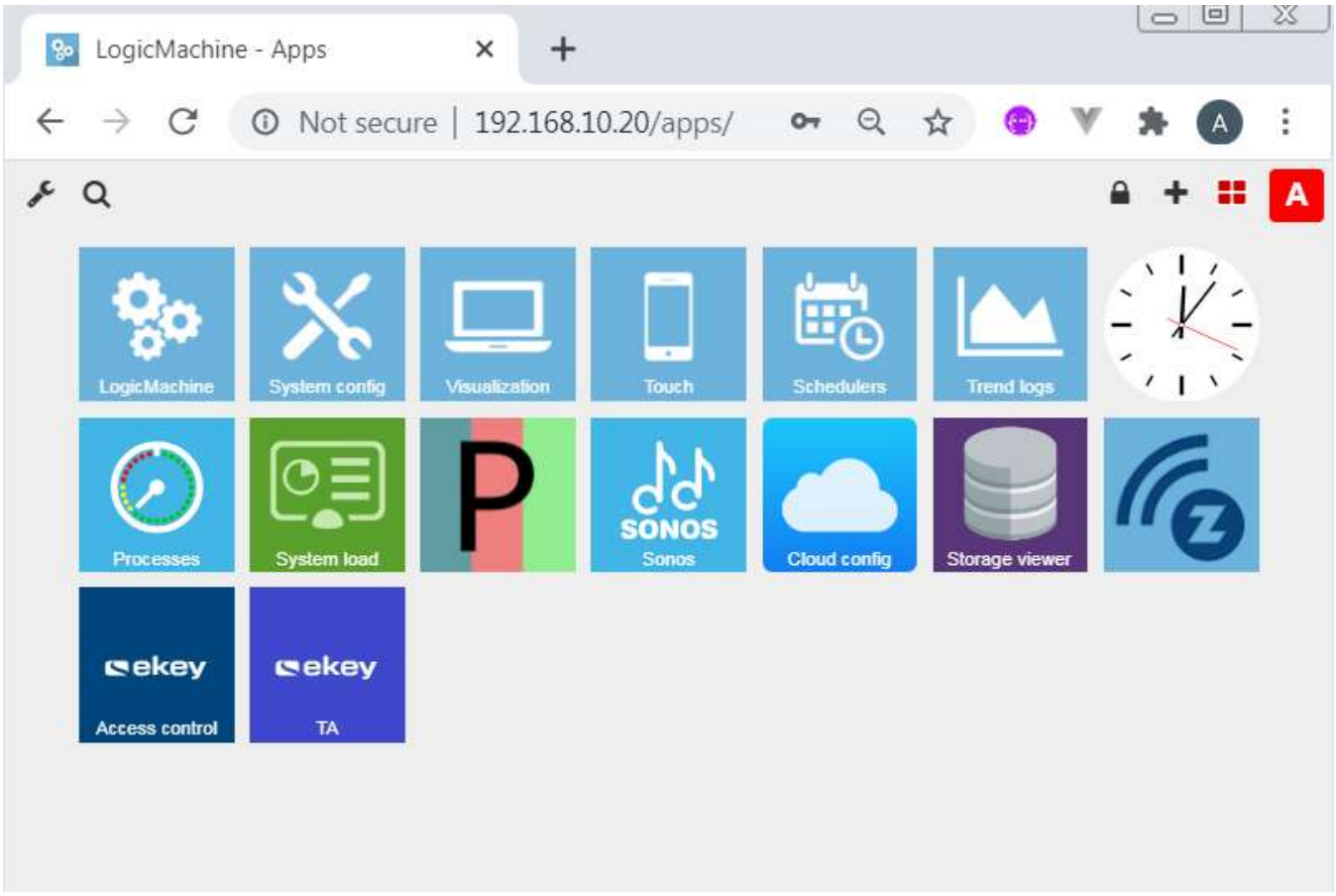
There are 2 types of LED:

- Status LED for operating status
- Function LED for indicating the function of the overall system.



1 Status LED
2 Function LEDs

5. Running the Application for the First Time



The active RS485 port name is registered into storage key `"app:ekey:port"`

When the active port is not known, the first run of the ekey app will search for the port within all available RS485 serial lines and register it as active when at least one ekey scanner is connected to this port.

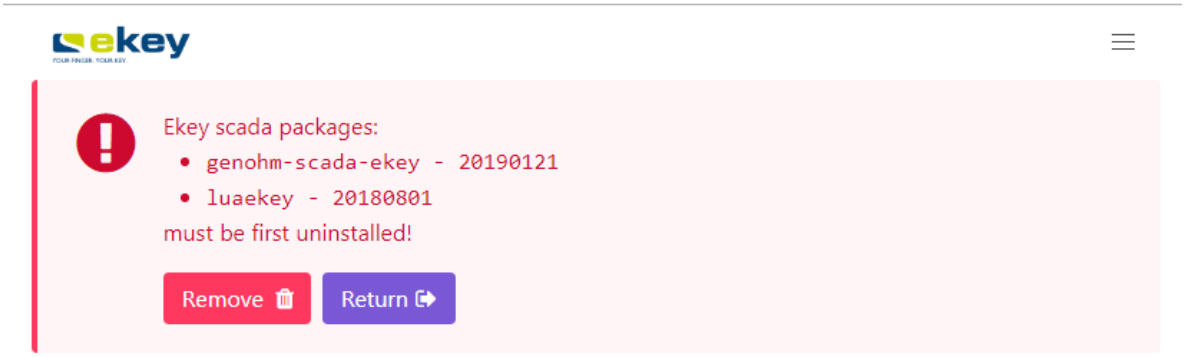
The sample view from the storage viewer app:

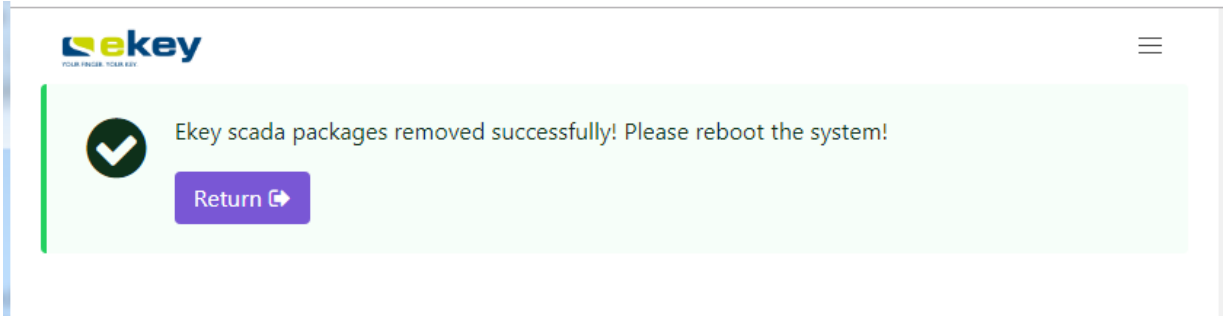
6. Migrating from the old “ekey scada LogicMacine application”

6.1. Removing the “ekey scada LogicMachine application”

If the “ekey scada LogicMacine application” is installed before, the first modal window will show you an option of uninstalling the corresponding packages.

You can press <Remove> to uninstall or press <Cancel> to exit the app. The existing database environment will not be deleted.
They will be used in this app after successful removal of ‘old ekey packages’. The existing lua scripts will continue to operate in this environment.





This process is reversable. If you later decide to install removed packages again, you can reinstall them again using the lua script below:

```
list = io.readproc ('opkg --force-depends install
http://www.ekey.lv/logicmachine/luaekey_20180801_imx6.ipk')
list = io.readproc('opkg --force-depends install http://www.ekey.lv/logicmachine/genohm-scada-
ekey_20190121_imx6.ipk')
```

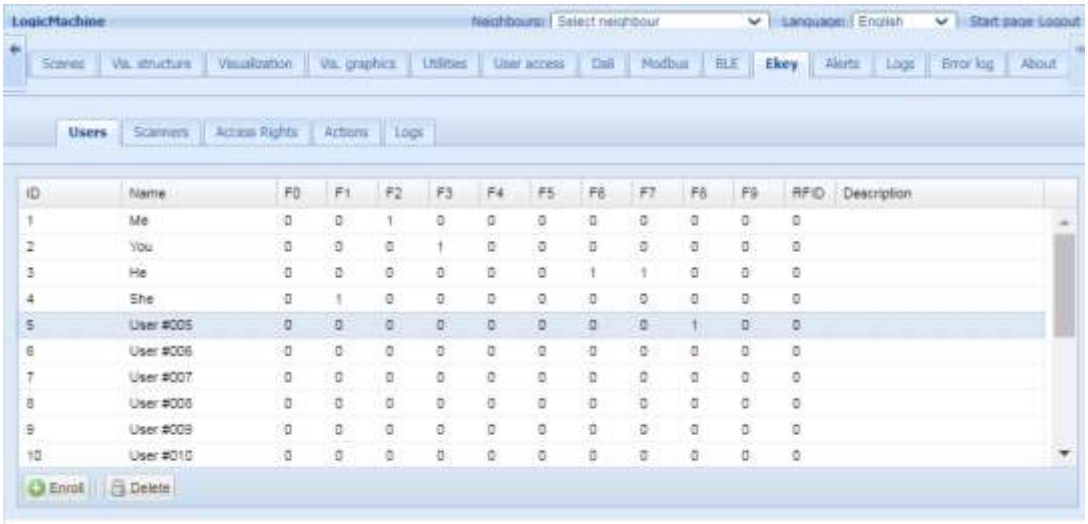
6.2. Installing the “ekey scada LogicMachine” as a 2nd front-end application

- This is a temporary solution because the LogicMachine will not support this method in the near future!

As an alternative, if you want to us the “ekey scada application” from LogicMachine tab “Ekey” with the same old look interface and without new futures from “ekey FSX app” you can copy the modifyed scada .js file to the right folder from lua script with this command:

```
os.execute('cp /home/apps/store/user/ekey/69-ekey-scada_app.js /www/scada/modules')
```

to get it visible again in the LogicMachine tab “Ekey”:

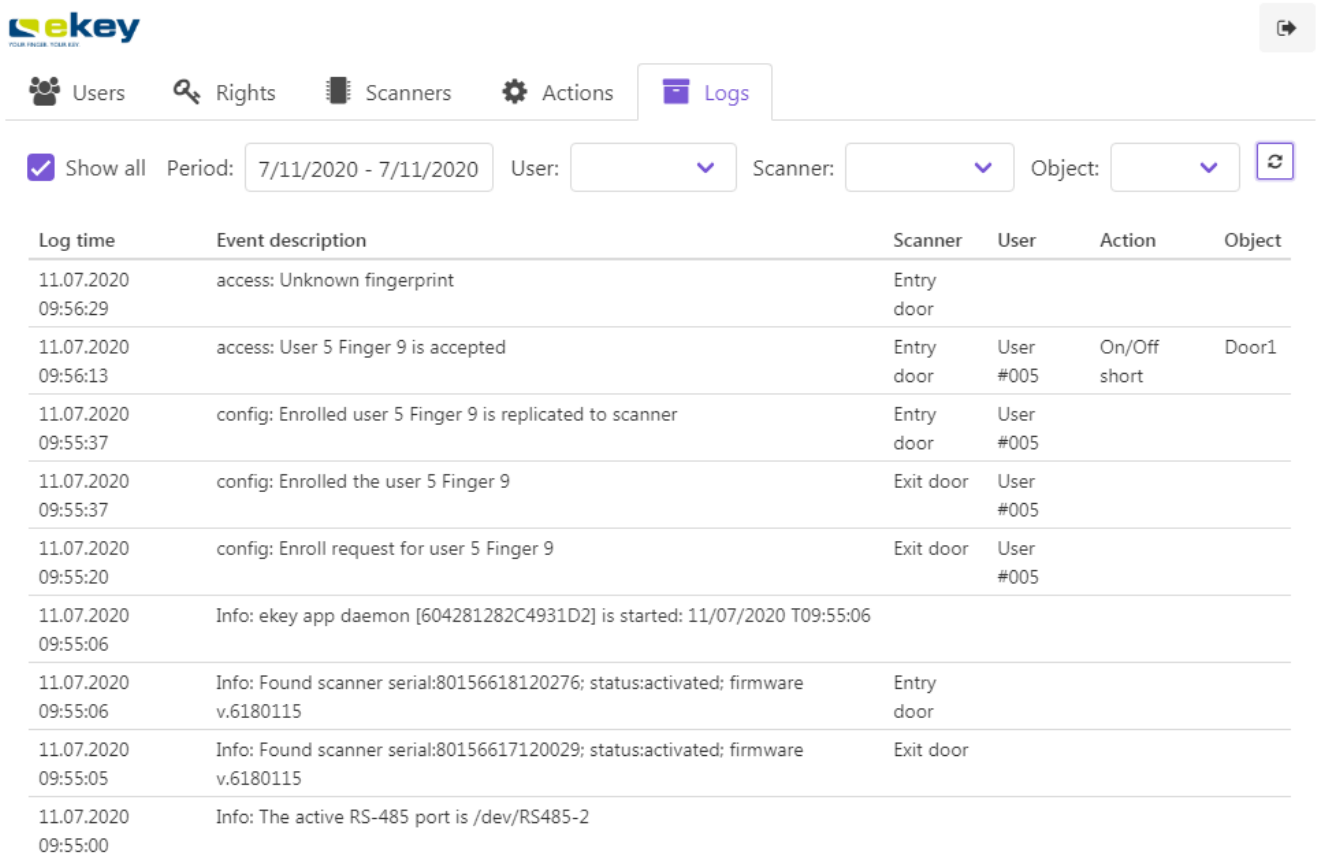


II. Ekey FSX App

The installed “ekey FSX app” is ready for use if you can see all 5 tabs:

1. Tab “Logs”

To view all events, incl. technical events, you need to set tickbox “Show all” to ‘Yes’.



- This screenshot shows the information where 2 scanners are connected to the RS485-2 line; have enrolled fingerprint 9 for User 5 and this fingerprint is replicated on the 2-nd scanner and has performed 2 access events : first positive with the same User 5 Finger 9 and the second negative on the second scanner.

Filter fields:

Show all - if “Yes” then include also additional technical Events (default value ‘No’)

Period: – Date from- Date to time interval (default value: today)

User: – Filter Log records only with selected User name

Scanner: – Filter Log records only with selected Scanner name

Object: – Filter Log records only with selected Object name

– command to refresh the list of records

Fields:

Log time – access or technical Event log data and time

Event description – detailed description of Event

Scanner – Scanner name at which Event occurs

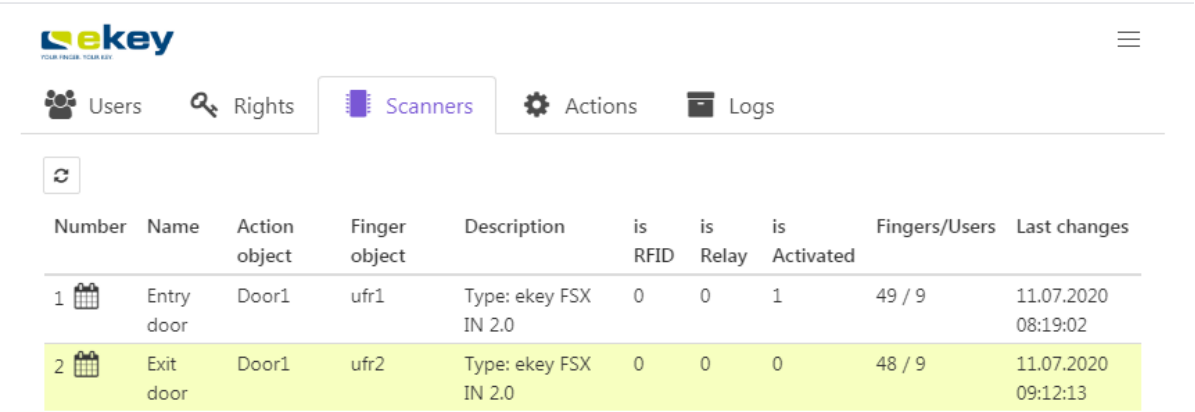
User – User name who created Event

Action – Action name which is initiated by Event

Object – KNX Object name on which Action is performed

2. Tab “Scanners”

The app can be operable if to the Logic Machine RS485 line are connected the ekey FSX type scanners.
A RS485 serial port can be choosed among other available Logic Machine ports.
* Alternatively the “ekey USB RS485 Converter” can be used to connect the scanners to the application.



Number	Name	Action object	Finger object	Description	is RFID	is Relay	is Activated	Fingers/Users	Last changes
1	Entry door	Door1	ufr1	Type: ekey FSX IN 2.0	0	0	1	49 / 9	11.07.2020 08:19:02
2	Exit door	Door1	ufr2	Type: ekey FSX IN 2.0	0	0	0	48 / 9	11.07.2020 09:12:13

The connected ekey FSX scanners automatically are added to the Scanners list.
The app recognizes the scanner type and creates its initial Description

Fields:

Number – Unique number.

The icon appears when the scanner ‘ is activated’ for Time attendance event recordings

Name – Scanner user short name

Action Object – KNX Data group address which will be activated after positive fingerprint recognition on this scanner performing the Action assigned to this User fingerprint.

Finger Object - KNX Data group address to which the information about the access event will be sent

Is RFID – the scanner property, if the scanner is prepared to use ekey RFID cards for access

Is Relay - the scanner property, if the scanner is equipped with a on-board relay

Is Activated – the application property for this scanner, which allows this scanner to be used without restrictions after 14 days of evaluation period within 3 months after production date.

The scanner record has a yellow background if the value “Is activated” = ‘No’.

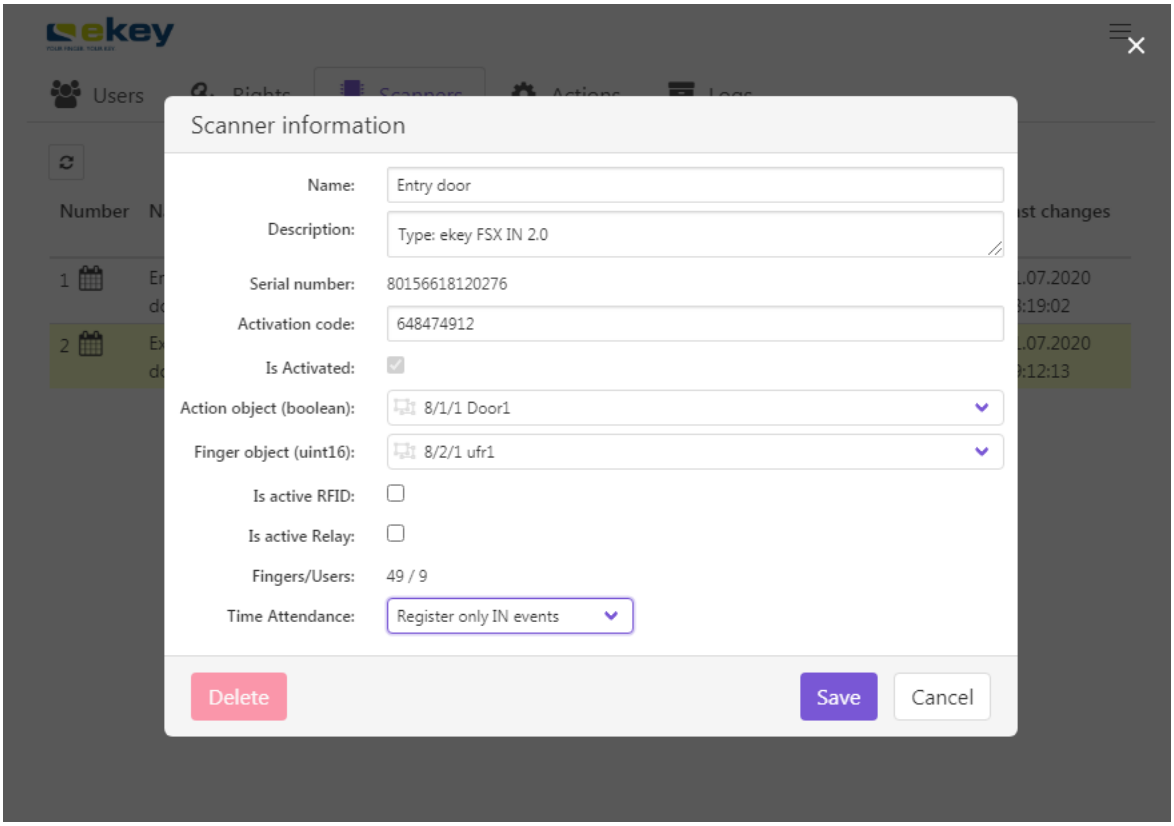
See detailed description: “Scanner information” field “Activation code”

Fingers/Users – the information from the scanner how many fingerprints there are recorded and from how many Users (A total of 200 fingerprints can be used in this system for each scanner)

Last changes – date and time of the last scanner’s record save event

A <**Click**> on the selected Scanner line opens a “Scanner information” card

a. Scanner information



Fields:

Name – Scanner name

Serial number – A serial number of the Scanner assigned by factory (read only)

Activation code – the code assigned to selected scanner for unlocking the full application functionality after 14 days of evaluation period within 3 months after production date.
The only limitation to use this scanner without a proper Activation code is not performing the Actions to this scanner assigned Objects after the positive fingerprint access event.

You must ask your ekey FSX scanner vendor for Activation keys.
When the ekey FSX fingerprint scanner is bought directly from the ekey app developer the Access code will be included in the scanner's price.
The Access code is transferable for this scanner between multiple ekey FS app installations.

Is Activated – the application property which allows the scanner to be used without restrictions after the evaluation period.

Action Object – KNX Data group address which will be activated after positive fingerprint recognition on this scanner performing the Action assigned to this User fingerprint.

Finger Object - KNX Data group address on which will be sent information about the access event (in form of …)

Is active RFID – the scanner property if the scanner is prepared for use ekey RFID cards

Is active Relay - the scanner property if the scanner is equipped with a on-board relay

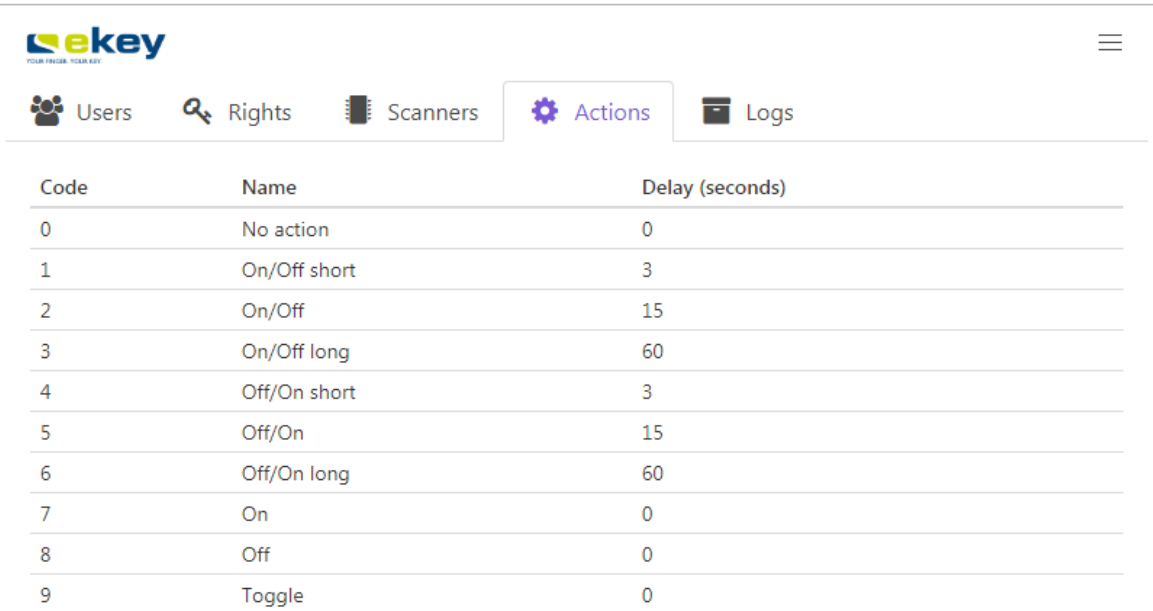
Fingers/Users – the information from the scanner how many fingerprints there are recorded and from how many Users (A total of 200 fingerprints can be used in this system for each scanner)

Time attendance – Time attendance event recording setting. User can select the values:
'No' (default), 'Register only IN events', 'Register only OUT events', 'Register IN and OUT events'.

Commands:

<**Delete**> - Delete the scanner record. This command is available when the scanner is not connected any more to the selected RS485 port line.
<**Save**> - Save the scanner data and close the window.
<**Cancel**> - reject the changes and close the window

3. Tab "Actions"



Code	Name	Delay (seconds)
0	No action	0
1	On/Off short	3
2	On/Off	15
3	On/Off long	60
4	Off/On short	3
5	Off/On	15
6	Off/On long	60
7	On	0
8	Off	0
9	Toggle	0

Fields:

Code – Action code

Name – Action name

Delay (seconds)– Action delay time in seconds (in case of the 2 sequential switching events are be performed)

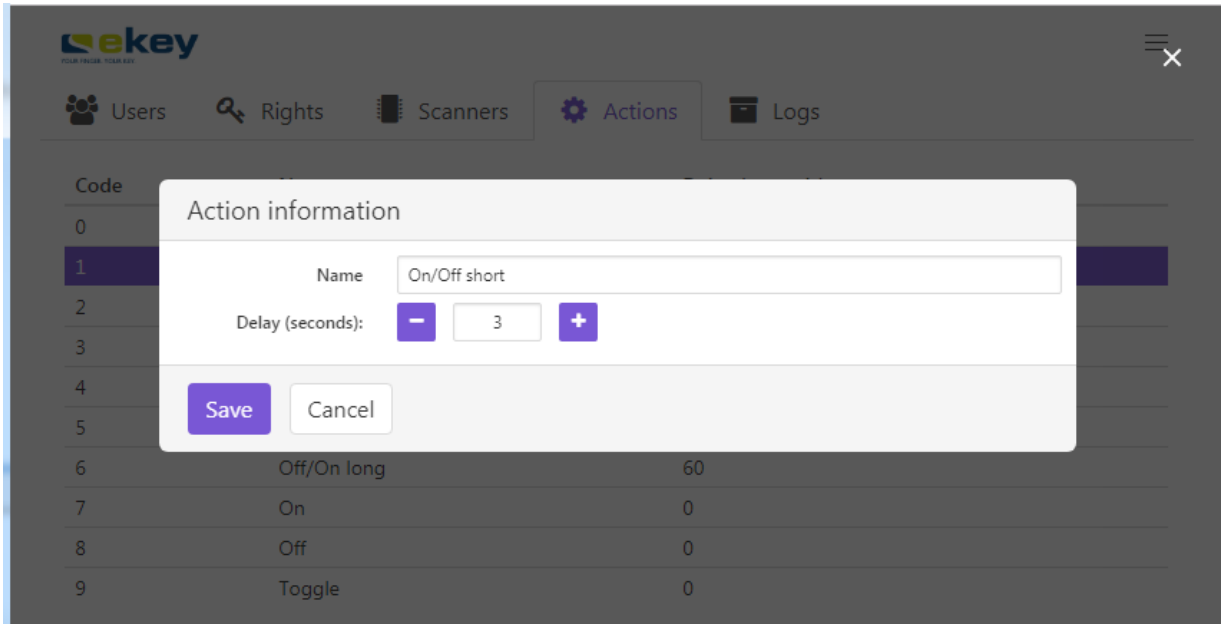
Records:

All Event records are predefined by the system. You can change only the Name and Delay values

Commands:

<**Click**> on the selected Action record line to open the “Action information” card

a. “Action information”



You can change only its Name and Delay values

Fields:

Name – Action name

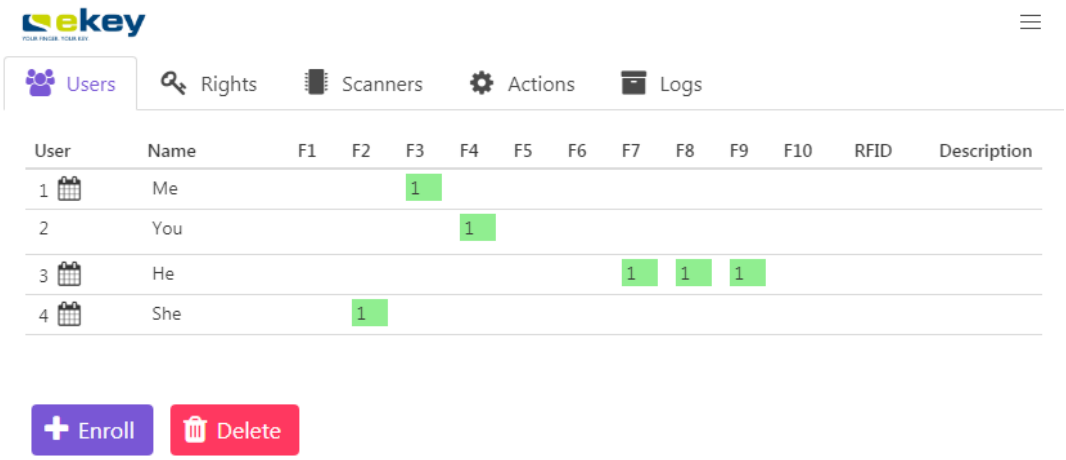
Delay (seconds)– Action delay time in seconds

Commands:

<**Save**> - Save the record data and close the Action card modal window.
<**Cancel**> - reject changes and close the Action card modal window

4. Tab “Users”

The system list contains the first 20 user records.
* If required the user list records can be expanded with a separate script to 200 records.
The columns from F1 to F10 and RFID show the Action code number assigned to the selected Users Fingers 0… 10 or RFID card accordingly.
The green background color shows that the User Finger is registered in the internal database of fingerprint credentials.



Fields:

User – User number

The icon appears when the user ‘ is checked’ for Time attendance event recordings

Name – User name

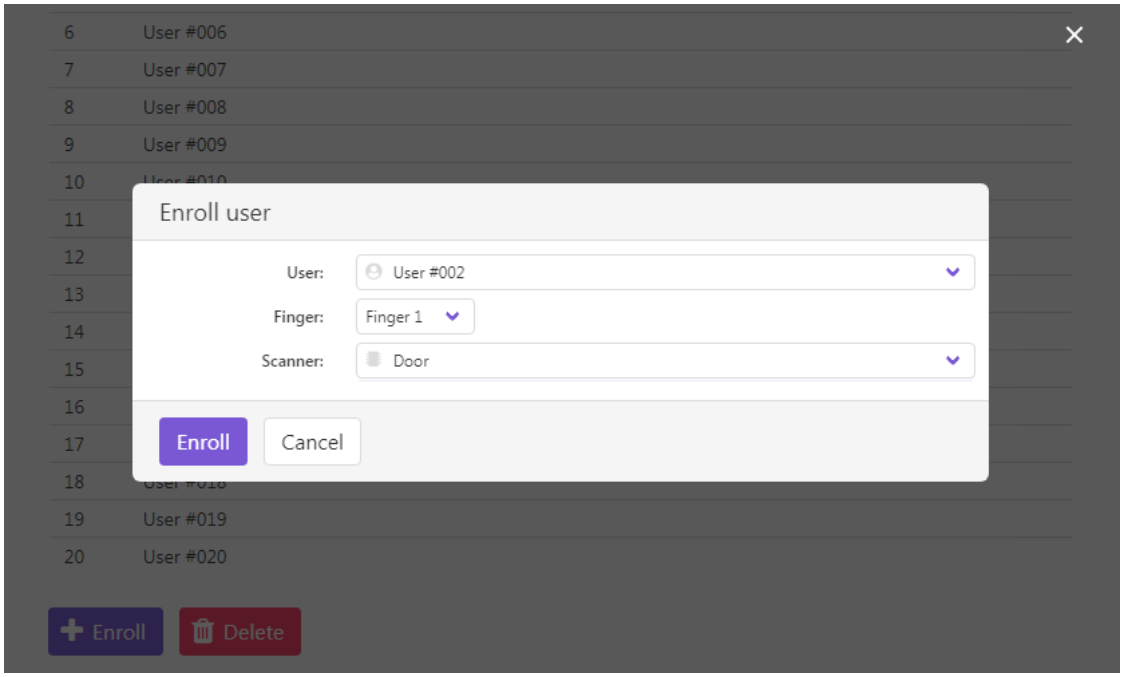
F1, .. , F10, RFID – An action number assigned to user fingers or RFID card

Description – Extra information about User.

Commands:

<**Click**> on the selected User line filed Name – Opens the **“User information”** card
<**Enroll**> - open the modal window “Enroll user”
<**Delete**> - open the modal window “Delete user”

a. “Enroll user”

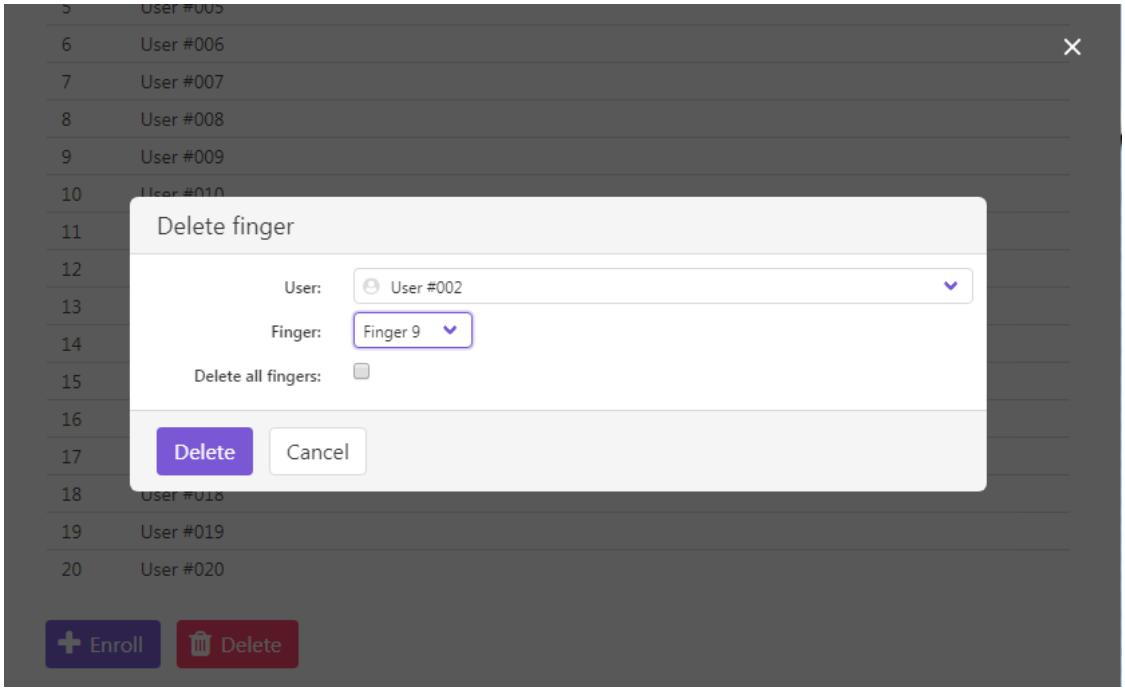


Commands:

<**Enroll**> - Enroll the selected User Fingerprint for the chosen ekey Scanner. The system will wait 30 sec to perform the operation.
* After successful enrollment you should refresh the User list to view the new information!

<**Cancel**> - close the modal window.

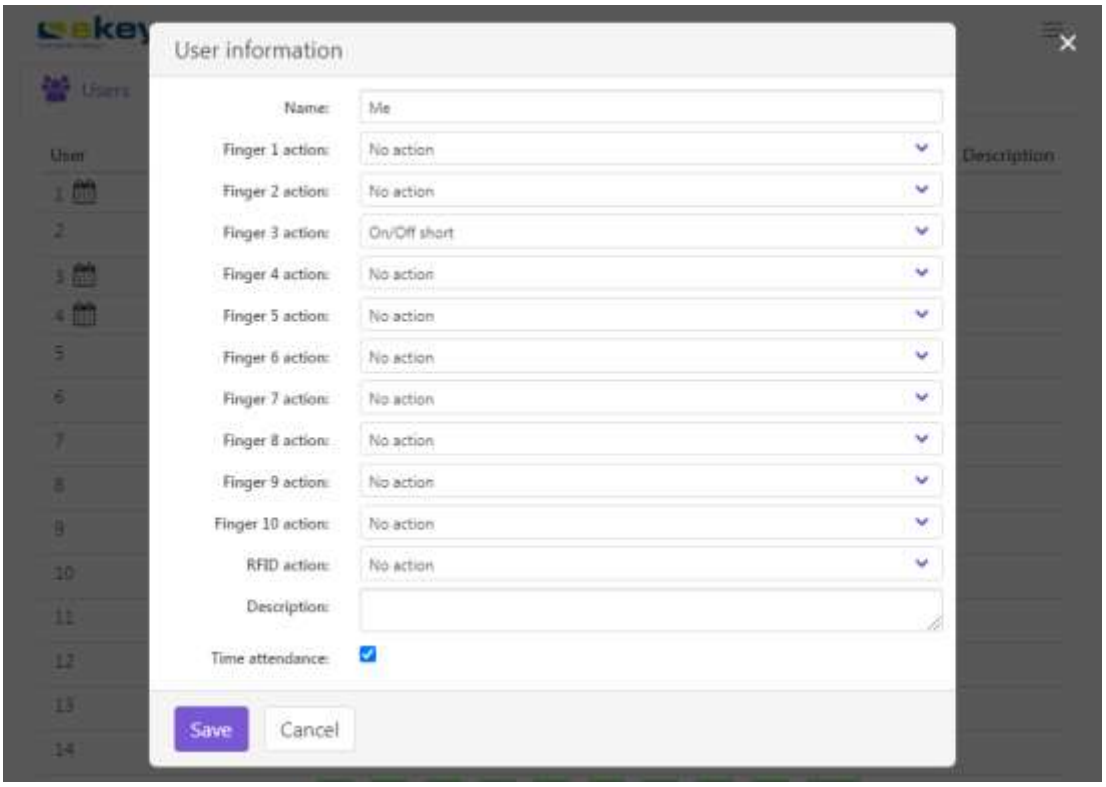
b. “Delete finger”



Commands:

<**Delete**> - Delete a selected User Finger or all fingers, if the “Delete all fingerprints” = ‘On’
<**Cancel**> - close the modal window.

c. “User information”



Fields:

Name – User name

Finger 1 action – selected Action name chosen for the selected user Finger 1

…

Finger 10 action – selected Action name chosen for the selected user Finger 2

RFID action – selected Action name chosen for the selected user RFID card

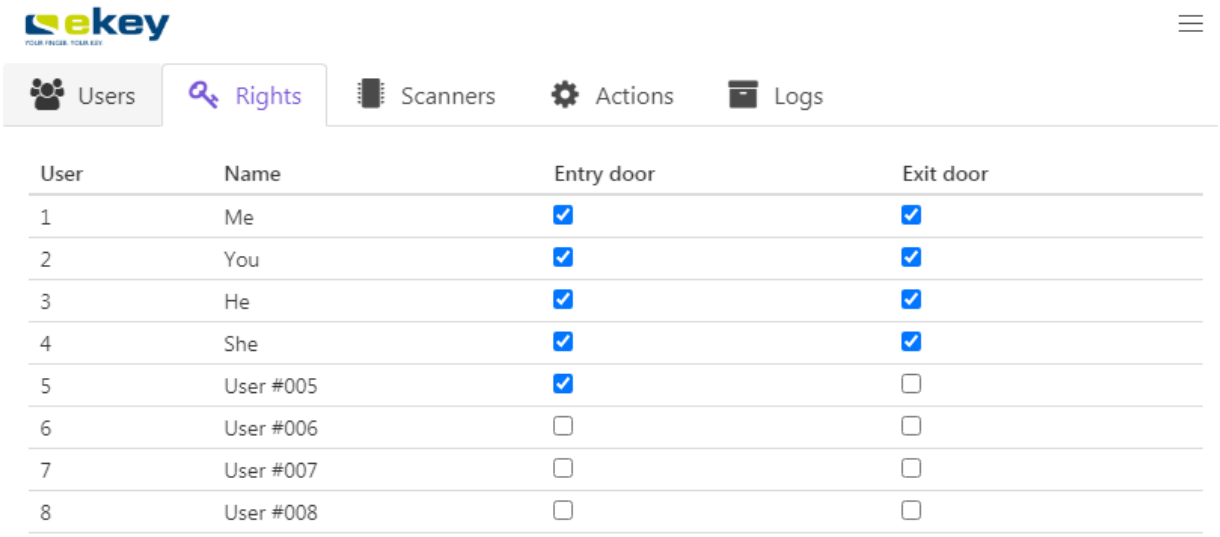
Description – Extra information about user.

Time attendance – the property to allow the User access events be recorded for Time attendance.

Commands:

<**Save**> - save the changes in fields
<**Cancel**> - close the modal window.

5. Tab "Rights"



Rows: The list contains the first 20 user records indicating a User number (Field User) and its name (field Name)
Columns: The next columns shows the scanner Name registered in the system.
Data: The cross-field value for a User (in Rows) and a Scanner (in Columns) can be either enabled or disabled by clicking on checkbox inside the column.

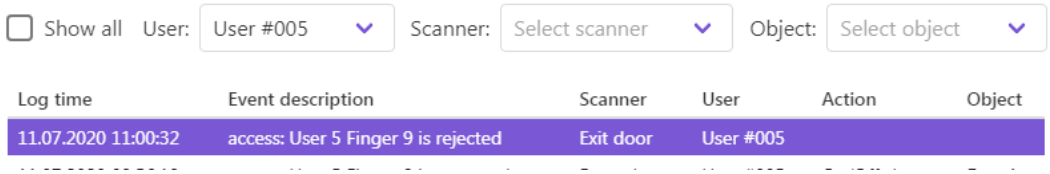
Commands:

<**Click**> on the field on/off between User row and Scanner column to enable/disable User rights for the selected scanner.

If the User has not access rights to the Scanner then

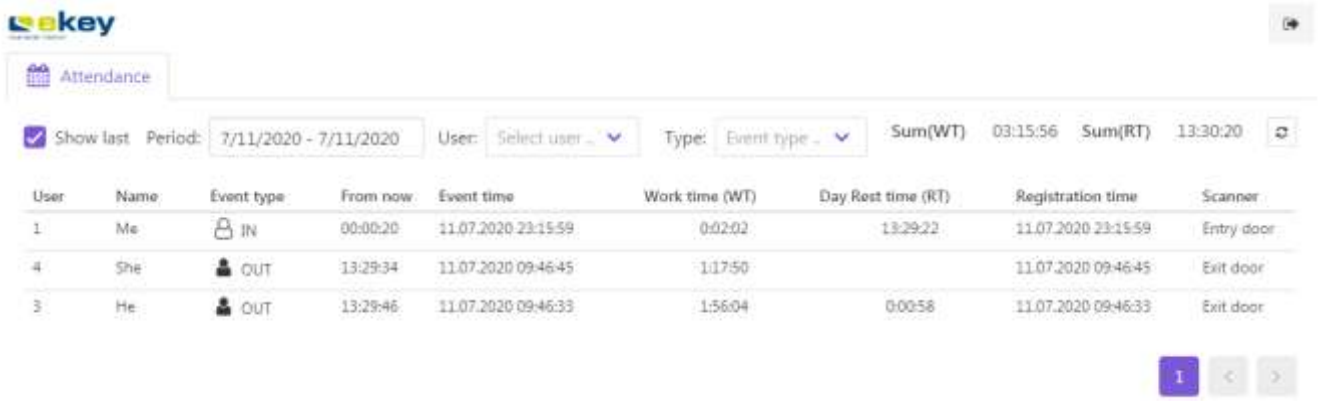
1) the scanner will recognize the user's fingerprint but will not perform the access event
2) The scanner status LED will light up green, but function LEDs will short light up red:

3) In the log file will be registered record " … is rejected"



III. Ekey TA App

6. Tab “Attendance”



Filter fields:

Show last - if “Yes” then the list wil show the summary information for each User with a “Time attendance” =’Yes’ property (default value ‘Yes’)

Period: – Date from- Date to time interval (default value: today)

User: – Filter Log records only with selected User name

Type: – Filter Log records only with selected Event types ‘ IN’ or ‘OUT’ (default value = both)

Sum(WT) – Sum of WorkTime (hh:mm:ss) for the filed “Work time (WT)”

Sum(RT) – Sum of RestTime (hh:mm:ss) for the filed “Day Rest time (WT)”

– command to refresh the list of records

Fields:

User – User number

Name – User name

Event type – ‘ IN’ event as Work time start condition or ‘ OUT’ as Work time end condition

From Now – Time elapsed from selection time

Event time – The datetime value for time attendance calculation

Work time (WT) – Work time period or sum of periods calculated as (Out time – IN time)

Day Rest time (RT) – Rest time period or sum of periods calculated as (IN time – OUT time) in the same calendar day

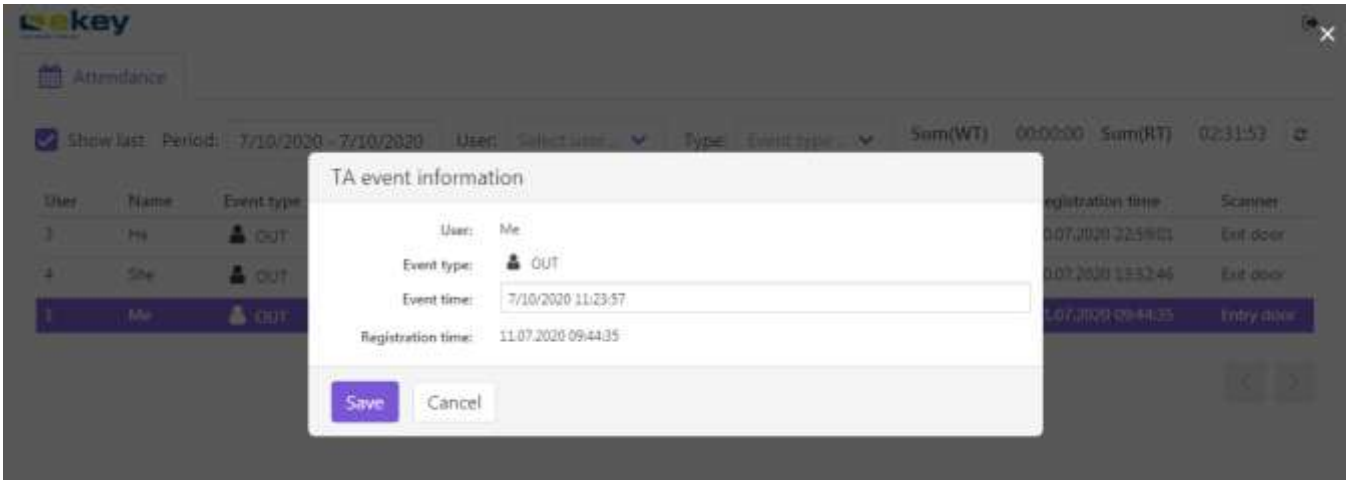
Registration time – The actual datetime value when the TA event was recorded

Scanner – Scanner name where the Time attendance event was recorded

Commands:

<Click> on the selected record field “Event time” or “Event type” will open the modal window “TA event information”

a. "TA event information"



Fields:

User: - User name /read only/

Event type: - Event type (IN or OUT) /read only/

Event time: - Event Time /editable value within time period between time interval from previous to next TA event Registration time /

Registration time - actual TA event registration time /read only/.

- If the Event time <> Registration time then the Event record will show the difference with the icon:

Commands:

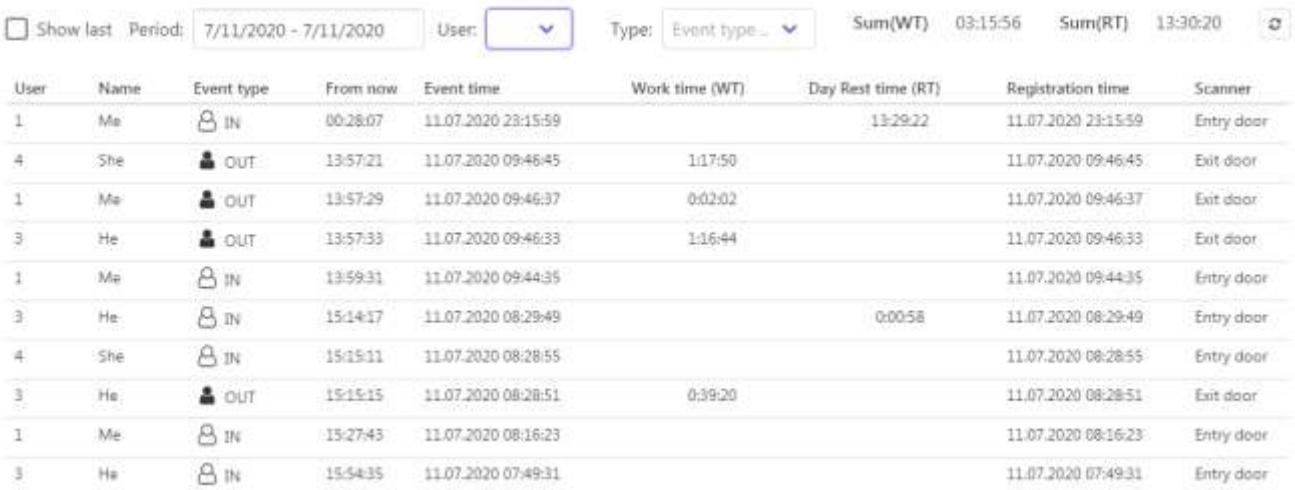
<**Save**> - Save the changes

<**Cancel**> - close the modal window.

Examples:

1. How to list all today TA events:

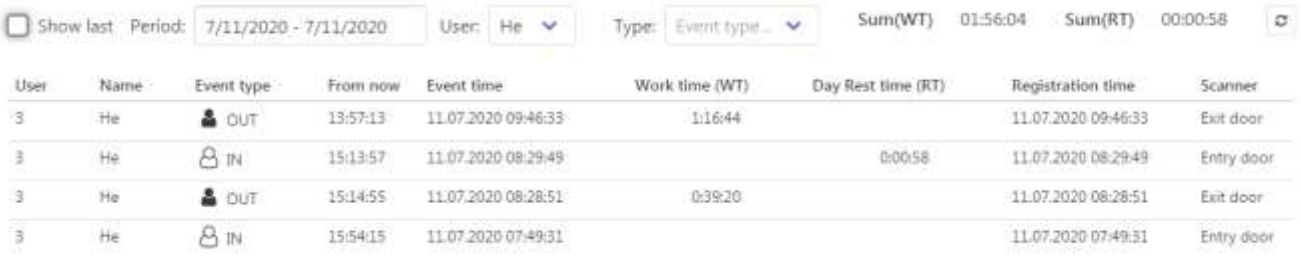
The Filter field “Show last” = ‘ No’



User	Name	Event type	From now	Event time	Work time (WT)	Day Rest time (RT)	Registration time	Scanner
1	Me	IN	00:28:07	11.07.2020 23:15:59		13:29:22	11.07.2020 23:15:59	Entry door
4	She	OUT	13:57:21	11.07.2020 09:46:45	1:17:50		11.07.2020 09:46:45	Exit door
1	Me	OUT	13:57:29	11.07.2020 09:46:37	0:02:02		11.07.2020 09:46:37	Exit door
3	He	OUT	13:57:33	11.07.2020 09:46:33	1:16:44		11.07.2020 09:46:33	Exit door
1	Me	IN	13:59:31	11.07.2020 09:44:35			11.07.2020 09:44:35	Entry door
3	He	IN	15:14:17	11.07.2020 08:29:49		0:00:58	11.07.2020 08:29:49	Entry door
4	She	IN	15:15:11	11.07.2020 08:28:55			11.07.2020 08:28:55	Entry door
3	He	OUT	15:15:15	11.07.2020 08:28:51	0:39:20		11.07.2020 08:28:51	Exit door
1	Me	IN	15:27:43	11.07.2020 08:16:23			11.07.2020 08:16:23	Entry door
3	He	IN	15:54:35	11.07.2020 07:49:31			11.07.2020 07:49:31	Entry door

2. How to list all today TA events for selected user :

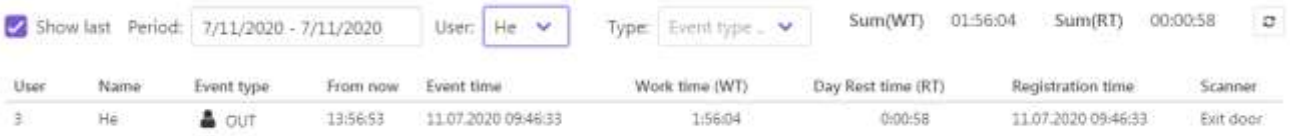
The Filter field “Show last” = ‘ No’ and User = ‘ He’



User	Name	Event type	From now	Event time	Work time (WT)	Day Rest time (RT)	Registration time	Scanner
3	He	OUT	13:57:13	11.07.2020 09:46:33	1:16:44		11.07.2020 09:46:33	Exit door
3	He	IN	15:13:57	11.07.2020 08:29:49		0:00:58	11.07.2020 08:29:49	Entry door
3	He	OUT	15:14:55	11.07.2020 08:28:51	0:39:20		11.07.2020 08:28:51	Exit door
3	He	IN	15:54:15	11.07.2020 07:49:31			11.07.2020 07:49:31	Entry door

3. How to list today’s summary information for selected user :

The Filter field “Show last” = ‘ Yes’ and User = ‘ He’



User	Name	Event type	From now	Event time	Work time (WT)	Day Rest time (RT)	Registration time	Scanner
3	He	OUT	13:56:53	11.07.2020 09:46:33	1:56:04	0:00:58	11.07.2020 09:46:33	Exit door

4. How to list today’s summary information for all user s:

The Filter field “Show last” = ‘ Yes’

User	Name	Event type	From now	Event time	Work time (WT)	Day Rest time (RT)	Registration time	Scanner
1	Me	IN	00:26:14	11.07.2020 23:15:59	0:02:02	13:29:22	11.07.2020 23:15:59	Entry door
4	She	OUT	13:55:28	11.07.2020 09:46:45	1:17:50		11.07.2020 09:46:45	Exit door
3	He	OUT	13:55:40	11.07.2020 09:46:33	1:56:04	0:00:58	11.07.2020 09:46:33	Exit door

The TA event registration:

I. When user register the new TA Event, the system always will show the feedback what type of event this is by short switching the scanner LED lights

<table>
<tr><td>IN event : start of the Work time period</td><td>OUT event: end of the Work time period</td></tr>
<tr><td colspan="2">1. Time attendance record registration:</td></tr>
<tr><td>  left function LED diode will switch green 0.3 seconds</td><td>  right function LED diode will switch red 0.3 seconds</td></tr>
<tr><td colspan="2">2. “Auto correction” of Time attendance record Time</td></tr>
</table>

<table>
<tr><td>Scanner “Time attendance” = ‘Register only IN events’</td><td>Scanner “Time attendance” = ‘Register only OUT events’</td><td>Scanner “Time attendance” = ‘Register IN and OUT events’</td></tr>
<tr><td>When the user register two sequential IN events then
a. the application will generate the missing previous OUT event with the Event Time = last known access event Registration Time before the duplicate IN event
b. left function LED diode will switch 2x green 0.3 seconds to indicate that the previous Work period time = 0h and it must be manually adjusted to correct value.</td><td>When the user register two sequential OUT events then
a. the application will generate the missing previous IN event with the Event Time = last known access event Registration Time after the previous OUT event;
b. right function LED diode will switch 2x red 0.3 seconds to indicate that the closed Work period time = 0h and it must be manually adjusted to correct value.</td><td>When the user register two sequential events within 60 second time interval (this an indication that user forget to register previous TA event) then
the previous TA event Time will be automatically corrected the same way as previous described when the user forget to register IN and OUT events.</td></tr>
</table>

3. “Manual correction” of Time attendance record Time

select the TA record you want to correct the Time value then <Click> on the record field “Event time” or “Event type” to open the modal window “TA event information” and change the “Event time” value to required.

IV. Other functions

a. Control scanners led lights (example: Event-based lua script for data group 8/1/5 Object alarm status)

Event-based: Show Object alarm status for Door1 scanner — Show code shortcuts

```
Alarm_data  = grp.find ('8/1/5')
Alarm = Alarm_data.data

  if Alarm then

     require('rpc').request('127.0.0.1', 8712, 'ekeyekeyekey', {
     cmd = 'ekey_fs_leds',
     colour = 0x1b    ,  -- Led light colour schema; default = 0x0d (off)
     time = 0xFFFFFFFF,  -- Duration of the required status in [ms]; default = 0xFFFFFFFF  (indefinitely)
     scanner = 1
  })

  else

    require('rpc').request('127.0.0.1', 8712, 'ekeyekeyekey', {
     cmd = 'ekey_fs_leds',
     colour = 0x0d    ,  -- Led light colour schema; default = 0x0d (off)
     time = 0xFFFFFFFF,  -- Duration of the required status in [ms]; default = 0xFFFFFFFF  (indefinitely)
     scanner = 1
  })

  end
```

Cmd description:

```
require('rpc').request('127.0.0.1', 8712, 'ekeyekeyekey', {
   cmd = 'ekey_fs_leds',
   colour = 0x1b    ,  -- red red ; default = 0x0d (off off)
   time = 0xFFFFFFFF,  -- Duration of the required status in [ms]; default = 0xFFFFFFFF  (indefinitely)
   scanner = 1
 })
```

Cmd ; LED left ; LED right

0x0d off off
0x0e green off
0x0f off green
0x10 green green
0x11 flashing green off
0x12 flashing green green
0x13 off flashing green
0x14 green flashing green
0x15 flashing green flashing green
0x16 * flashing green flashing green
0x17 ** flashing green flashing green
0x18 *** flashing green flashing green
0x19 red off
0x1a off red
0x1b red red
0x1c yellow off
0x1d off yellow
0x1e yellow yellow
0x1f green red
0x20 green yellow
0x21 red green
0x22 yellow green
0x23 red yellow

0x24 yellow red

b. Control the scanner onboard-relay

```
require('rpc').request('127.0.0.1', 8712, 'ekeyekeyekey', {
   cmd = 'ekey_fs_relay',
   scanner = 1 ,    --  selected scanner by ID value;
   command = 2 ,    --  switch type command: 0 = Off; 1=On; 2=impulss;   Default = 0
   relayID = 1 ,    --  scanner relay ID ; Default = 1 (the scanner onboard-relay)
   impulss = 3000   --  duration of the required status in [ms] or default: 3000ms
})
```

c. Script to decode user/finger pair (fingers start at 0, users at 1)

-- The positive access Event information sent to Scanners assigned “Finger object” can be decoded :

```
value = event.getvalue()
user = bit.rshift(value, 4)
finger = bit.band(value, 0x0F)
local direction = tonumber(require('dbenv'):new():getone('SELECT direction FROM ekey_TA_events WHERE id in (select max(id)
FROM ekey_TA_events WHERE user_id=?)', user)) or 2
--  Time attendance ‘direction’  value decoding pattern {{value = 0, key ='Exit'},{value = 1, key = 'Entry' }, {value = 2, key = 'no information'}}
local username = require('dbenv'):new():getone('SELECT name FROM ekey_users WHERE id = ?', user)
alert('User: %d, Finger: %d, Direction: %d, Username: %s' ,  user, finger, direction, username)
```

d. Service commands to manage the scanner fingerprints

1) Clean the scanner fingerprint/RFID memory:
http://192.168.0.10/apps/data/ekey/api.lp?request=**scanner-clean**&scanner=2&deviceid= 604281282C4931D2

2) Copy the fingerprints from the internal database to selected scanner
http://192.168.0.10/apps/data/ekey/api.lp?request=**copy-from-db-to-scanner**&scanner=2&deviceid=604281282C4931D2

3) Copy the fingerprints from the selected scanner to internal database
http://192.168.0.10/apps/data/ekey/api.lp?request=**copy-from-scanner-to-db**&scanner=2&deviceid=604281282C4931D2

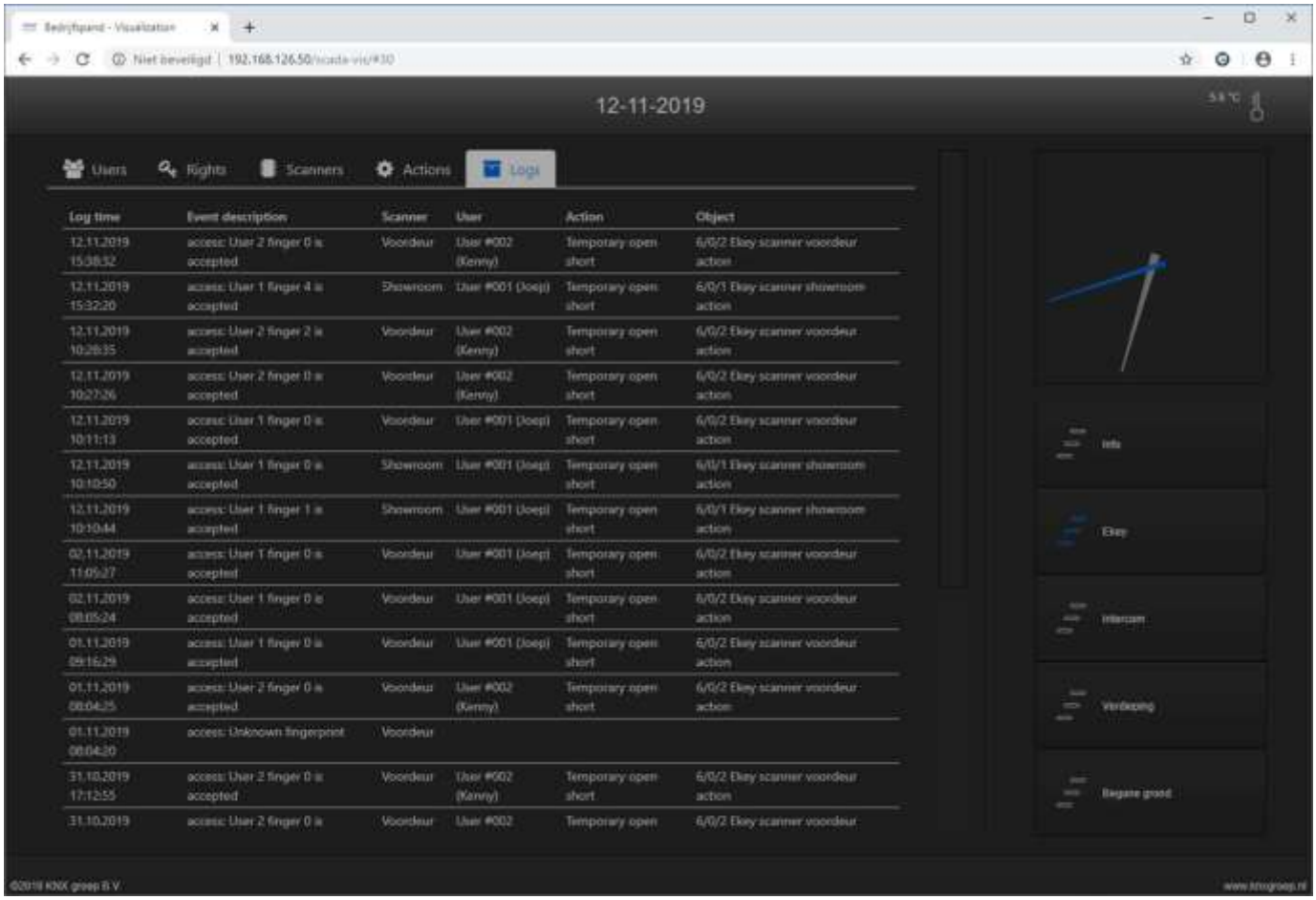
all 3 commands require 2 parameters:
- “scanner” - the selected scanner filed “Number” value from the tab “Scanners”
- "deviceid" - the LogicMachine device identification number

the deviceid value can contain [8 or 16 simbols] and is visible in the log record when the ekey FSX app daemon was started:

```
11.07.2020      Info: ekey app daemon [604281282C4931D2] is started:
09:54:35        11/07/2020 T09:54:35
```

e. ekey app visual design options

The ekey FSX app initial visual design can be transformed to specific client needs, like this example received from KNX groep B.V. *https://knxgroep.nl*



V. Feedback:

- Questions and suggestions please send to email: agris@avu.lv