Project Identification	
Project number	AAL-2013-6-064
Duration	1st July 2014 until 30th June 2017
Coordinator	Martin Biallas
Coordinator Organisation	Lucerne University of Applied Sciences and Arts – Engineering & Architecture, CEESAR-iHomeLab, Horw, Switzerland
Website	www.TransSafe.eu

Requirements and Specification for WebUI/API Database PrivacyAndDataProtection

Document Identification	
Deliverable ID:	Addendum_D4.3 Requirements and Specification for WebUI/API Database PrivacyAndDataProtection
Release number/date	V2.0 25.02.2016
Checked and released by	Urs Müller, KON
Work Status	Work in Progress
Review Status	Not reviewed

Key Information from "Description of Work"	
Deliverable Description	Requirements and specification for Design, Web API, Web UI, Privacy and data protection related to the Trans.Safe database. This document is updated continuously.

Dissemination Level	Confidential
Deliverable Type	Other
Original due date	25.02.2016

Authorship& Reviewer Information	
Editor	Carmine Troiano KON
Partners contributing	YOUSE, HSLU, TIL, VAG, MAN
Reviewed by	Urs Müller KON

Release History

Release Number	Date	Author(s)	Release description /changes made Please make sure that the text you enter here is a brief summary of what was actually changed; do not just repeat information from the other columns.
V 01	25.02.2016	Carmine Troiano	First version. I gathered the four documents “Requirements and Specification for Web UI” “Requirements and Specification for Web API” “Requirements and Specification for Privacy and Data Protection” and “Requirements and Specification for Database Design” into this Document. The other four documents are obsolete
V 02	25.02.2016	Carmine Troiano	add chapter Communication Security

Trans.Safe Consortium

Trans.Safe (AAL-2013-6-064.) is a project within the AAL Joint Programme Call 6

The consortium members are:

Partner 1	**Lucerne University of Applied Sciences and Arts – Engineering & Architecture, CEESAR-iHomeLab (Project Coordinator, HSL, CH)**
Contact person:	Martin Biallas
Email:	Martin.Biallas@hslu.ch

Partner 2	**Youse GmbH (YOU, DE)**
Contact person:	Cornelia Schauber
Email:	Cornelia.Schauber@youse.de

Partner 3	**Telecom Italia S.p.A. (TIL, IT)**
Contact person:	Gianluca De Petris
Email:	Gianluca.dePetris@telecomitalia.it

Partner 4	**VAG Verkehrs-AG Nürnberg (VAG, DE)**
Contact person:	Andreas May
Email:	Andreas.May@vag.de

Partner 5	**MAN Truck & Bus AG (MAN, DE)**
Contact person:	Walter Schwertberger
Email:	Walter.Schwertberger@man.eu

Partner 6	**Scuola Superiore Sant' Anna (SSSA, IT)**
Contact person:	Filippo Cavallo
Email:	F.Cavallo@sssup.it

Partner 7	**konplan systemhaus ag (KON, CH)**
Contact person:	Andy Tonazzi
Email:	Andy.Tonazzi@konplan.com

Partner 8	**Design LED Products Ltd (DLED, UK)**
Contact person:	James Gourlay
Email:	James.Gourlay@designledproducts.com

Table of Contents

Table of Contents
Table of Figures
Abbreviations
1 Overview
1.1 System Description Summary
1.2 Scope
2 Web UI Requirements
2.1 Introduction
2.2 Non-Functional Requirements
2.3 Functional Requirements
2.3.1 User Roles and Rights
2.3.2 General View Requirements
2.3.3 View Log in
2.3.4 View Home Page
2.3.5 View User management
2.3.6 View System Status
2.3.7 External Interfaces
2.3.7.1 Connection to Gateways / Sensors / Devices
2.3.7.2 REST Interface for mobile app client
2.3.8 Error Handling
2.3.8.1 General Error Handling
3 Web UI Requirement collection for future
3.1 User Roles and Rights
3.1.1 Admin Add / Edit / Remove Proband or Analyser
3.2 User Login
3.3 Gateway / Device management
3.3.1 Status of Gateways
3.3.2 Data statistics
3.3.2.1 Wearable Device Self-Registration
3.3.2.2 Proband Assign Device
3.3.2.3 Admin Add / Edit / Remove Device
3.3.3 Stress Observation Trigger Intervention / Notify Proband
3.4 Error Handling
3.4.1 Logging
3.4.2 Wearable Sensor Activity
3.4.3 Environmental Sensor Activity
3.5 Technology
4 Web API Functions
4.1 Version
4.2 Requests
4.2.1 Token
4.2.2 Delete data

4.2.3 Systemlog
4.2.4 Devicelist
4.2.5 Interventionlist
4.2.6 Intervention
4.2.7 stressdata
4.3 Status Codes
5 Database Design
5.1 Version 2016.02.25
6 Communication Security
6.1 TLS
6.2 HSTS
6.3 HPKP

Table of Figures

Figure 1: Trans.Safe Architecture Overview... 1

Abbreviations

Abbrev.	**Description**
UI	User Interface
MVVM	Model - View - Viewmodel
API	Application Programming Interface
GW	Gateway
ORM	Object Relational Mapping
MVC	Model View Controller
REST	Representational State Transfer
SQL	Structured Query Language
RAML	RESTful API Modeling Language
TLS	Transport Layer Security
HSTS	HTTP Strict Transport Security
HPKP	Public Key Pinning Extention for HTTP
HTTP(S)	HyperText Transfer Protocol (Secure)

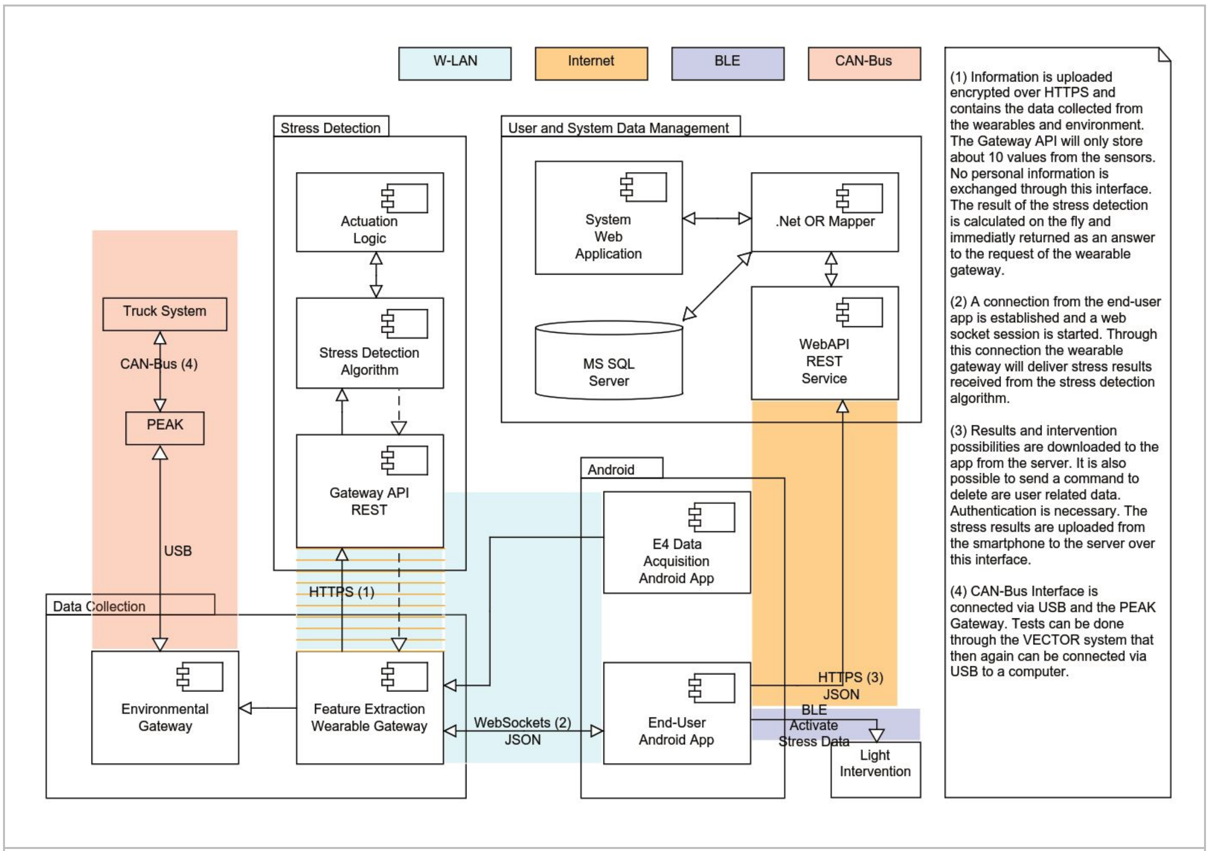
1 Overview

1.1 System Description Summary

The Trans.Safe systems purpose is to monitor workers and provide them with feedback or direct interventions on their overload or underload. From that point of view, the systems main tasks are

- Collect relevant data of workers
- Collect relevant environmental data
- Decide on overload/underload
- Inform user or start intervention

Therefore, the system is designed into several functional parts, which are further described in the following chapters.



(1) Information is uploaded encrypted over HTTPS and contains the data collected from the wearables and environment. The Gateway API will only store about 10 values from the sensors. No personal information is exchanged through this interface. The result of the stress detection is calculated on the fly and immediatly returned as an answer to the request of the wearable gateway.

(2) A connection from the end-user app is established and a web socket session is started. Through this connection the wearable gateway will deliver stress results received from the stress detection algorithm.

(3) Results and intervention possibilities are downloaded to the app from the server. It is also possible to send a command to delete are user related data. Authentication is necessary. The stress results are uploaded from the smartphone to the server over this interface.

(4) CAN-Bus Interface is connected via USB and the PEAK Gateway. Tests can be done through the VECTOR system that then again can be connected via USB to a computer.

Figure 1: Trans.Safe Architecture Overview

The Cloud Service System consists of two parts “Stress Detection” and “User & System Data Management”. This document defines the specifications for the subsystem “User & System Data Management”.

1. The subsystem “Stress Detection” is responsible for receiving the gateway data, managing gateway connection, doing the actual stress detection and handling interventions.
2. The subsystem “User & System Data Management” is responsible for user handling, system management and phone app “End-User Android App“ connection.

1.2 Scope

This document holds the functional description of the Web UI, Web API. It also contains a Database diagram and Privacy and Data Protection.

2 Web UI Requirements

2.1 Introduction

This chapter describes the functional and non functional requirements of the Trans.Safe systems Web UI.

2.2 Non-Functional Requirements

N°	Description
FSUI-100	The System must be able to handle up to 100 active users.
FSUI-110	The intervention data must be available at least one month.
FSUI-120	It must be possible to maintain the system infrastructure using remote desktop technology.

2.3 Functional Requirements

2.3.1 User Roles and Rights

<table>
<tr><th>N°</th><th>Description</th></tr>
<tr><td>FSUI-130</td><td>The system has three (3) different user roles.<ul><li><b>Administrator</b> (User with the all rights, e.g. changing configuration, adding users)</li><li><b>Analyzer</b> (User with the right to view all recorded (aggregated) data of probands)</li><li><b>Proband</b> (Normal user of the system. Monitored user, for which data is collected and stress detection takes place)</li></ul></td></tr>
<tr><td>FSUI-140</td><td>A user with the role Proband is not allowed to login via the web UI</td></tr>
<tr><td>FSUI-150</td><td>A user with the role Proband can only login via smartphone app</td></tr>
<tr><td>FSUI-160</td><td>A user role has the following rights in the system:<table><tr><th>Rights</th><th>Logged out
(Level 0)</th><th>Proband
(Level 1)</th><th>Analyzer
(Level 2)</th><th>Administrator
(Level 3)</th></tr></table></td></tr>
</table>

Different pages and buttons accessible from all pages				
Login page	x			
Can Loggin to WebUI HomePage			x	x
TransSafe Main button	x		x	x
About button enabled	x		x	x
Contact button enabled	x		x	x
Change Password button enabled (not visible from Login page)			x	x
Log out button enabled (visible while user is logged in)			x	x
Log In button enabled (visible while not logged in)	x			
WebUI Home Page				
Logged In Home page			x	x
User Management button visible			x	x
User Management button enabled				x
System Status button visible			x	x
System Status button enabled				x
Data Statistics button visible			x	x
Data Statistics button enabled				
Change Password button visible and enabled			x	x
User Management page				
List of users				x
Add User button enabled				x
Edit button enabled				x
Info button enabled				x
Delete button enabled				x
Back to Home button enabled				x
System Status page				

Log List view				x
Filter view				x
Back to Home button enabled				x

2.3.2 General View Requirements

N°	Description
FSUI-170	The Header Bar and the Footer is shown on every View
FSUI-180	The Header Bar contains the following 5 buttons: ● Trans.Safe ● About ● Contact ● Hello xxx! ● Log off/in
FSUI-190	By pressing on the Header Bar button Trans.Safe the view Home Page shall be shown. Only when no user is logged in the Log In view is shown.
FSUI-200	By pressing on the Header Bar button About the view About shall be shown.
FSUI-210	By pressing on the Header Bar button Contact the view Contact shall be shown.
FSUI-220	By pressing on the Header Bar button Hallo xxx! the view Change Password shall be shown.
FSUI-230	The button Log off/in depends if a user is logged in or not: ● User Logged in: The name of the button is Log off. By pressing on the Header Bar button Log off the view Log In shall be shown. ● User Logged off: The name of the button is Log in. By pressing on the Header Bar button Log in the view Log In shall be shown.
FSUI-240	By pressing on the Header Bar button Log off the view Log in shall be shown.
FSUI-250	The Footer shall contain the version of WEB UI and the copy right text "@xxxx - Trans.Safe" (xxxx is the year of copyright)
FSUI-260	Logout is possible from any page.

2.3.3 View Log in

N°	Description
FSUI-270	When calling the Trans.Safe Cloud System only the login page including the header bar and footer is displayed. ● No menu is shown ● No further system information is shown
FSUI-280	The Log in view shall have the entry fields “User” and “Password”
FSUI-290	The Log in view shall have a check box “Remember me?”
FSUI-300	The Log in view shall have the button “Log in”.
FSUI-310	By pressing on the button “Log in” the system shall validate the user name and password, if it is correct it shall show the view Home Page. If the validation failed the system shall inform the user why the validation failed by showing a text.
FSUI-320	System events like user has been logged in where stored in a dedicated database table

2.3.4 View Home Page

N°	Description
FSUI-330	The view Home Page has the following five buttons: ● User Management ● Device Management ● Gateway Management ● System Status ● Data Statistics
FSUI-340	Logout is possible from any page.

2.3.5 View User management

N°	Description
FSUI-350	The Main User Management view shall show a User List with all users
FSUI-360	The User List has the following column: ● User Name ● Email ● Phone Number

	- Action
FSUI-370	The column Action of the User List has the following three buttons: - Edit - Info - Delete
FSUI-380	The Main User Management view shall have a Add User button. By pressing on it the Add User view shall be shown.
FSUI-390	The Main User Management view shall have a “Back to Home” button. By pressing on it the Home view shall be shown.

2.3.6 View System Status

N°	Description
FSUI-400	The System Status view shall show a Log List.
FSUI-410	The Log List shall have the following information per log entry: - DateTime stamp - Status Level - Event description
FSUI-420	The System Status view shall have a Filter for the Log List
FSUI-430	The Log List Filter shall have the following settings to filter: - Date From - Date To - Status Level
FSUI-440	The Log List Filter shall have a Filter button. By pressing on it the Log List shall be updated according the filter settings (only changing the filter settings without pressing the button, is not updating the Log List).

2.3.7 External Interfaces

2.3.7.1 Connection to Gateways / Sensors / Devices

N°	Description
FSUI-450	The web UI can not connect to any gateway directly
FSUI-460	The web UI can not connect to any sensor directly
FSUI-470	The web UI can not connect to any device directly

2.3.7.2 REST Interface for mobile app client

The REST Interface for mobile app client is defined in [2].

2.3.8 Error Handling

2.3.8.1 General Error Handling

N°	Description
FSUI-480	General Error messages (stack traces, SQL Server, system messages) are not displayed to proband users or analyser users
FSUI-490	General Error messages are written to the Web UI Log.

3 Web UI Requirement collection for future

3.1 User Roles and Rights

<table>
<tr><th>N°</th><th>Description</th></tr>
<tr><td>FSUI-500</td><td>The system has the following user groups and rights
<table>
<tr><th>Rights</th><th>Logged out (Level 0)</th><th>Proband (Level 1)</th><th>Analyzer (Level 2)</th><th>Administrator (Level 3)</th></tr>
<tr><td colspan="5"><b>Gateway/Device Management pages</b></td></tr>
<tr><td>(Add)/Edit/Remove wearable device</td><td></td><td></td><td></td><td>x</td></tr>
<tr><td>(Add)/Edit/Remove gateway</td><td></td><td></td><td></td><td>x</td></tr>
<tr><td>Register mobile app device</td><td></td><td>x</td><td></td><td></td></tr>
<tr><td>Register wearable device</td><td></td><td></td><td></td><td></td></tr>
<tr><td>Back to Home button enabled</td><td></td><td></td><td></td><td>x</td></tr>
<tr><td colspan="5"><b>Data Statistic page</b></td></tr>
<tr><td>Alter recorded wearable/environmental data</td><td></td><td></td><td></td><td></td></tr>
<tr><td>View recorded (aggregated) data and stress detection statistics for whole system</td><td></td><td></td><td>x</td><td>x</td></tr>
<tr><td>View recorded personal data and stress detection statistics</td><td></td><td></td><td></td><td></td></tr>
<tr><td>Back to Home button enabled</td><td></td><td></td><td></td><td>x</td></tr>
<tr><td colspan="5"><b>Handy App Access</b></td></tr>
<tr><td>Login Page</td><td>x</td><td></td><td></td><td></td></tr>
<tr><td>Can Login</td><td></td><td>x</td><td>???</td><td>???</td></tr>
</table>
</td></tr>
<tr><td>FSUI-510</td><td>Proband can't request for a new password in case of forget or lost it</td></tr>
</table>

3.1.1 Admin Add / Edit / Remove Proband or Analyser

N°	Description
FSUI-720	Adding or removing proband, analyser or another administrator must only be possible for an admin user
FSUI-730	Adding an analyser can be deactivated upon installation
FSUI-740	A proband, for which data has been collected, must not be deletable, but may only be set to inactive
FSUI-750	For a proband, the following information is going to be stored in the database belonging to the system: - Username - Password
FSUI-760	For a analyser, the following information is going to be stored in the system: - Username - Password
FSUI-770	No personal data that is required by the stress algorithm can be entered in the web UI
FSUI-780	Recorded data (raw) and personal data is not visible in the web UI

3.2 User Login

N°	Description
FSUI-630	Last system login is recorded and displayed in the personal data page.

3.3 Gateway / Device management

N°	Description
FSUI-520	The system offers three (3) types of hardware devices that can be managed ● Gateway for environmental sensors ● Gateway for wearable devices ● Wearable device
FSUI-530	The types of wearable and environmental devices can be managed on a specific administration page in the web user interface

FSUI-540	Gateways for environmental sensors and gateways for wearables are added manually via according web UI management pages.
FSUI-550	For a wearable device, the following information is stored in the database: ● Device Id (Identification ID in the database) ● Identifier (a name identifier) ● Serialnumber ● Manufacturer ● Description
FSUI-560	For a gateway (environmental), the following information is stored in the database: ● Gateway Id (Identification ID in the database) ● Identifier (a name identifier) ● Modified data ● Description ● Last Time Data received
FSUI-570	For a gateway (wearable), the following information is stored in the database: ● Gateway Id ● Identifier ● Modified data ● Description ● Last Time Data received

3.3.1 Status of Gateways

N°	Description
FSUI-580	The status of gateway for wearables and gateway for environmental sensors is displayed for administration users only
FSUI-590	The status of wearable sensors is displayed for administration users only

3.3.2 Data statistics

N°	Description
FSUI-600	Statistical data can be viewed by administrators and data analysers
FSUI-610	Statistical data: Average number of interventions in a time period
FSUI-620	Time period for average calculation can be chosen by entering start- and end date or suitable user controls

3.3.2.1 Wearable Device Self-Registration

N°	Description
FSUI-790	When a wearable device is providing data to the database via the gateway for the first time, the wearable device is added to the system database. The device is made identifiable by an unique identifier. The unique identifier cannot be changed.

3.3.2.2 Proband Assign Device

N°	Description
FSUI-800	Assigning a wearable sensor to a proband via the web UI is not possible

3.3.2.3 Admin Add / Edit / Remove Device

N°	Description
FSUI-640	For each device type, the system provides an own entry in the main menu.
FSUI-650	For each device type, an overview page containing all currently registered devices is available, from which a details page for an individual device can be show.
FSUI-660	Adding or removing a device is only possible for Admin user.
FSUI-670	If user data has been connected to a specific device serial number, the device must not be deletable, but may only be set to inactive.
FSUI-680	For environmental sensor data, the following information is stored in the system: ● Gateway Id ● Type ● Value ● Unit ● Time stamp
FSUI-690	For wearable sensor data, the following information is stored in the system: ● Gateway Id ● Type ● Value ● Unit ● Time stamp ● Device Id

FSUI-700	The status for environmental gateway and gateway for wearables are the following: ● **Active** (Added to system, sending data) ● **Inactive** (Added to system, not sending data, no tracker registered)
FSUI-710	A wearable sensor may have the following states: ● **Active** (Added to system, assigned to user, sending data) ● **Inactive** (Added to system, data has been recorded, not in usage any more)

3.3.3 Stress Observation Trigger Intervention / Notify Proband

N°	Description
FSUI-810	If an intervention happens, the system must inform the users Mobile App client via push or poll mechanism

3.4 Error Handling

3.4.1 Logging

N°	Description
FSUI-820	Errors regarding web application exceptions are logged in the database

3.4.2 Wearable Sensor Activity

N°	Description
FSUI-830	The website lists a wearable sensor as active, if the last time data received is lower than 10 minutes.

3.4.3 Environmental Sensor Activity

N°	Description
FSUI-840	The environmental gateway sensors are not listed or treated separately. The gateway and its sensors are a single entity.
FSUI-850	The website list an environmental gateway as active, if the last time data received is lower than 10 minutes.

3.5 Technology

N°	Description
FSUI-860	Visual Studio 2013 is used as development tool
FSUI-870	.Net and according C# language Version 4.5 are used
FSUI-880	Entity Framework 6 is used to implement the data access part of the User & System Management Part of the system
FSUI-890	ASP.Net MVC 5 is used for the web part of the User & System Management part
FSUI-900	ASP.Net Web API 2 is used to implement the REST interface for external applications, e.g. the mobile app
FSUI-910	Microsoft SQL Server 2014 is used as the cloud systems database

4 Web API Functions

4.1 Version

The Web Api is versioned to provide backward compatibility. You can choose between a version by changing the url path accordingly. For example change .../v1/... to .../v2/… to use WebApi version 2.

4.2 Requests

4.2.1 Token

Authenticate the user with the system and obtain the auth_token

Request

Method	URL
POST	/token

Header

Params	Type	Example
Accept	string	application/json
Content-Type	string	application/x-www-form-urlencoded

Body

Params	Type	Example

grant_type
username
password
	string
	string
	string
		grant_type=password
		username=username
		password=password

Example: grant_type=password&username=username&password=password

Response

Status	Response
200	{"access_token": "0DW29xUqtg2AOBlhU7ExLbHXQfGp3Jpx0TfpvSXppoS5JG7TDTiitaV4hQzjqs Zdd7U5HfWKQvKTHR020bc98V-ZmtmkbdSsk1fWprXflzQyKnjsp9N8sSCm-Q8 0eqmNNP-T8pCJmHWyziSss_qwETYIXeh0AWwsOeUbEQ0mamrJqwTPSBE- h0wPbfKxbhP28hrAlbhFgrL4ExbsNo09LqpBbl8tq7XA4MsvDh6XmAElIzP5iYLy Ooh1dwVvV9", "token_type": "bearer", "expires_in": 86399 ".issued": "Fri, 11 Sep 2015 10:18:36 GMT", ".expires": "Sat, 12 Sep 2015 10:18:36 GMT" }
400	{"error":"invalid_grant","error_description":"The user name or password is incorrect."}
500	{"message":"An error has occurred.","exceptionMessage"}

4.2.2 Delete data

Delete user specific data using a valid token

Request

Method	URL
POST	/deletedata

Header

Params	Type	Example
Authorization	string	Bearer 0DW29xUqt...
Accept	string	application/json
Content-Type	string	application/json

Response

Status	Response
200	{ status : "success", data : null }
400	{"message":"Validation failed for one or more entities. See 'EntityValidationErrors' property for more details."}
401	{"message":"Authorization has been denied for this request."}
405	{"message":"The requested resource does not support http method."}
500	{"message":"An error has occurred.","exceptionMessage"}

4.2.3 Systemlog

Insert system information to the database using a valid token

Request

Method	URL
POST	/systemlog

Header

Params	Type	Example
Authorization	string	Bearer 0DW29xUqt...
Accept	string	application/json
Content-Type	string	application/json

Body

Params	Type	Example
LogMessage	string	Custom Log Message
LogLevel	int	1:Trace 2:Info 3:Warn 4:Error
LogDate	date	2015-09-08T13:03:43

Example: **{ LogMessage :"Log Message",LogLevel:"1", LogDate:"2015-09-08T13:03:43" }**

Response

Status	Response
200	{ status : "success", data : null }
400	{"message":"Validation failed for one or more entities. See 'EntityValidationErrors' property for more details."}
401	{"message":"Authorization has been denied for this request."}
405	{"message":"The requested resource does not support http method."}
500	{"message":"An error has occurred.","exceptionMessage"}

4.2.4 Devicelist

Returns a list of all devices by using a valid token

Request

Method	URL
GET	/devicelist

Header

Params	Type	Example
Authorization	string	Bearer 0DW29xUqt...
Accept	string	application/json
Content-Type	string	application/json

Response

Status	Response

200	{
	"DeviceId":1,"DeviceIdentifier":"device1","Manufacturer":"siemens","Used":0,"Description":null,"DeviceType":null},{"DeviceId":2,"DeviceIdentifier":"device2","Manufacturer":"q-tec","Used":1,"Description":null,"DeviceType":null
	}

400	{"message":"Validation failed for one or more entities. See 'EntityValidationErrors' property for more details."}
401	{"message":"Authorization has been denied for this request."}
405	{"message":"The requested resource does not support http method."}
500	{"message":"An error has occurred.","exceptionMessage"}

4.2.5 Interventionlist

Returns a list of interventions using a valid token

Request

Method	URL
GET	/interventionlist

Header

Params	Type	Example
Authorization	string	Bearer 0DW29xUqt...
Accept	string	application/json
Content-Type	string	application/json

Response

Status	Response
200	{"InterventionTypeId":1,"InterventionTypeName":"Light","InterventionTypeDes ription":"Light shower / Light intervention"}, {"InterventionTypeId":2, "InterventionTypeName":"Feedback","InterventionTypeDescription":"Feedbac / Workload"},{"InterventionTypeId":3,"InterventionTypeName":"Quiz","Interventi nTypeDescription":"Answer questions"}, {"InterventionTypeId":4, "InterventionTypeName":"Breathing","InterventionTypeDescription":"Breathing exercise"},{"InterventionTypeId":5,"InterventionTypeName":"Moving","Interven onTypeDescription":"Movement exercise"}, {"InterventionTypeId":6, "InterventionTypeName":"Assistance","InterventionTypeDescription":"Technic l (driver) assistance systems"}
400	{"message":"Validation failed for one or more entities. See 'EntityValidationErrors' property for more details."}
401	{"message":"Authorization has been denied for this request."}
405	{"message":"The requested resource does not support http method."}
500	{"message":"An error has occurred.","exceptionMessage"}

4.2.6 Intervention

Returns user interventions (possible, favorite, rated) for a specific user using a valid token. Post taken/selected intervention for a specific user using a valid token.

Request

Method	URL
GET/POST	/intervention

Header

Params	Type	Example
Authorization	string	Bearer 0DW29xUqt...
Accept	string	application/json
Content-Type	string	application/json

Body

Params	Type	Example
InterventionReason	string	Intervention reason text.
InterventionType	int	1-6
InterventionDate	date	2015-09-08T13:03:43

Response

Status	Response

200	[
	  {
	        "InterventionPrefsId": 1,
	        "InterventionRating": 0,
	        "InterventionSelected": true,
	        "InterventionType": null
	  },
	  {
	        "InterventionPrefsId": 2,
	        "InterventionRating": 0,
	        "InterventionSelected": true,
	        "InterventionType": null
	  },

	{
	"InterventionPrefsId": 3,
	"InterventionRating": 0,
	"InterventionSelected": false,
	"InterventionType": null
	},
	{
	"InterventionPrefsId": 4,
	"InterventionRating": 0,
	"InterventionSelected": false,
	"InterventionType": null
	},
	{
	"InterventionPrefsId": 5,
	"InterventionRating": 0,
	"InterventionSelected": false,
	"InterventionType": null
	},
	{
	"InterventionPrefsId": 6,
	"InterventionRating": 0,
	"InterventionSelected": false,
	"InterventionType": null
	}
	]

400	{"message":"Validation failed for one or more entities. See 'EntityValidationErrors' property for more details."}
401	{"message":"Authorization has been denied for this request."}
405	{"message":"The requested resource does not support http method."}
500	{"message":"An error has occurred.","exceptionMessage"}

4.2.7 stressdata

Returns detection results related to specific user using a valid token. Insert detection result related to specific user using a valid token.

Request

Method	URL
GET/POST	*/stressdata?f=01.01.2015%201:20:20%20am&t=31.12.2015%201:20:20%20am*

Header

Params	Type	Example
Authorization	string	Bearer 0DW29xUqt...
Accept	string	application/json
Content-Type	string	application/json

Body

Params	Type	Example
DoIntervention	bool	True/False
Level	double	1.0 to 10.0
DetectionTime	date	2015-09-08T13:03:43

Response

Status	Response
200	{ status : "success", data : null }
400	{"message":"Validation failed for one or more entities. See 'EntityValidationErrors' property for more details."}
401	{"message":"Authorization has been denied for this request."}
405	{"message":"The requested resource does not support http method."}
500	{"message":"An error has occurred.","exceptionMessage"}

4.3 Status Codes

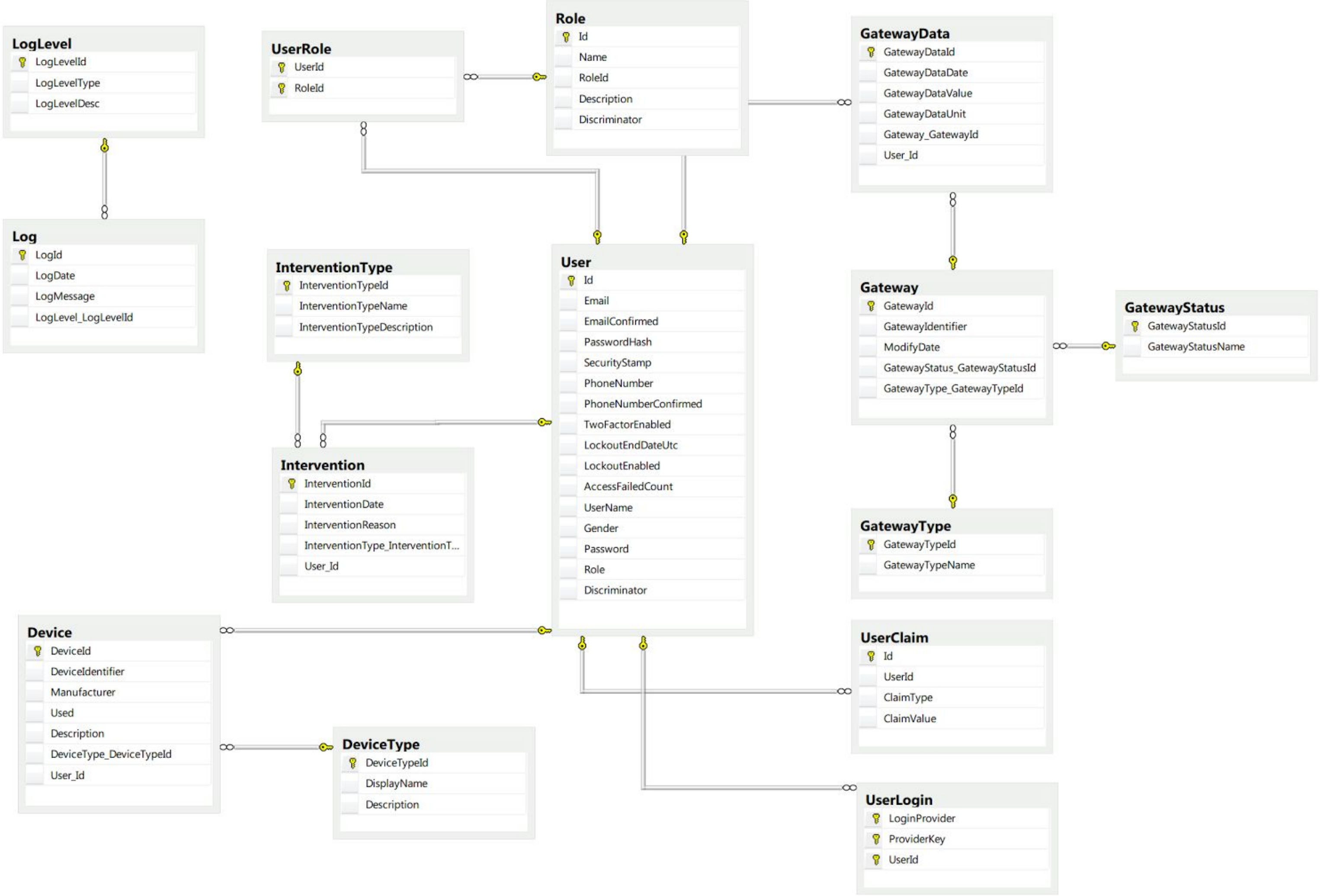
All status codes are standard HTTP status codes. The below ones are used in this API.

2XX - Success of some kind
4XX - Error occurred in client's part
5XX - Error occurred in server's part

Status Code	Description
200	OK
201	Created
202	Accepted (Request accepted, and queued for execution)
400	Bad request
401	Authentication failure
403	Forbidden
404	Resource not found
405	Method Not Allowed
409	Conflict
412	Precondition Failed
413	Request Entity Too Large
500	Internal Server Error
501	Not Implemented
503	Service Unavailable

5 Database Design

5.1 Version 2016.02.25



6 Communication Security

6.1 TLS

Transport Layer Security offers the following security features:

- server authentication
- integrity protection
- peplay protection
- confidentiality
- X509 Certificates

6.2 HSTS

HTTP Strict Transport Security is a security feature that lets a web site tell browsers that it should only be communicated with using HTTPS, instead of using HTTP.
How it helps

If a web site accepts a connection through HTTP and redirects to HTTPS, the user in this case may initially talk to the non-encrypted version of the site before being redirected, if, for example, the user types http://www.foo.com/ or even just foo.com.

This opens up the potential for a man-in-the-middle attack, where the redirect could be exploited to direct a user to a malicious site instead of the secure version of the original page.

The HTTP Strict Transport Security feature lets a web site inform the browser that it should never load the site using HTTP, and should automatically convert all attempts to access the site using HTTP to HTTPS requests instead.

6.3 HPKP

To ensure the authenticity of a server's public key used in TLS sessions, this public key is wrapped into a X.509 certificate which is usually signed by a certificate authority (CA). Web clients such as browsers trust a lot of these CAs, which can all create certificates for arbitrary domain names. If an attacker is able to compromise a single CA, he can perform MITM attacks on various TLS connections. HPKP can circumvent this threat for the HTTPS protocol by telling the client which public key belongs to a certain web server.

HPKP is a Trust on First Use (TOFU) technique. The first time a web server tells a client via a special HTTP header which public keys belong to it, the client stores this information for a given period of time. When the client visits the server again, it expects a certificate containing a public key whose fingerprint is already known via HPKP. If the server delivers an unknown public key, the client should present a warning to the user.