D6.3 PLATFORM INITIAL PROTOTYPE

Developing the platform and its modules

Project title	**Collaborative Recommendations and Adaptive Control for Personalised Energy Saving**
Project acronym	**enCOMPASS**
Project call	**EE-07-2016-2017 Behavioural change toward energy efficiency through ICT**
Work Package	**WP6**
Lead Partner	**SMOB**
Contributing Partner(s)	**PMI, CERTH, GRA, PDX, SUPSI**
Security classification	**Public**
Contractual delivery date	**30/04/2018**
Actual delivery date	**30/04/2018**
Version	**1.0**
Reviewers	**CERTH, SUPSI**

History of changes

Version	Date	Comments	Main Authors
0.1	01/02/2018	Defining Table of Content	L.Caldararu (SMOB)
0.2	16/03/2018	Defining DDP	L.Caldararu (SMOB)
0.21	19/03/2018	Validating DDP	P.Fraternali (PMI)
0.25	05/04/2018	Added enCOMPASS Database installation notes Added Smart Meter and Sensor Data Management component installation description	L.Caldararu (SMOB) G.Cristea (SMOB)
0.3	09/04/2018	Added Gamification Engine installation description Added Awareness Applications installation description	S.Herrera, C.Pasini (PMI)
0.4	09/04/2018	Added Recommendation Engine installation description	G.Vadasz (GRA)
0.5	12/04/2018	Updated of Recommendation Engine installation notes	G.Vadasz (GRA)
0.6	16/04/2018	Added Energy Efficiency Console Platform for Building Managers installation description	S.Krinidis (CERTH)
0.7	18/04/2018	Added Service Integration and Component Orchestration (Semaphorization) installation description Added Notification Engine installation description	L.Caldararu (SMOB) G.Cristea (SMOB)
0.9	20/04/2018	Compiled content from all Partners	L.Caldararu (SMOB)
0.91	21/04/2018	Indexed Tables and Figures Updated Notification Engine installation description	L.Caldararu (SMOB)
0.92	25/04/2018	First internal review	S. Krinidis (CERTH)
0.93	27/04/2018	Second internal review	A.E. Rizzoli (SUPSI)
1.0	30/04/2018	Final quality check	P. Fraternali (PMI)

Disclaimer

This document contains confidential information in the form of the enCOMPASS project findings, work and products and its use is strictly regulated by the enCOMPASS Consortium Agreement and by Contract no. 723059.

Neither the enCOMPASS Consortium nor any of its officers, employees or agents shall be responsible or liable in negligence or otherwise howsoever in respect of any inaccuracy or omission herein.

The contents of this document are the sole responsibility of the enCOMPASS consortium and can in no way be taken to reflect the views of the European Union.

***This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 723059*.**

TABLE OF CONTENTS

enCOMPASS platform – Intial Prototype

Table of Figures

Table of Tables

EXECUTIVE SUMMARY

This is the **accompanying document** of the Deliverable **D6.3: Platform Initial prototype**, which, according to the Description of Action (DoA), is:

Software deliverable of the initial prototype of the enCOMPASS platform. It provides the infrastructure to collect energy consumption data, first integration of the user interfaces designed in WP2, for early testing in the pilots. Source code and installation documentation

Therefore this document describes a **software deliverable** containing the initial prototype of the enCOMPASS platform. It provides the infrastructure to collect and organize the energy consumption data as well as data reported by temperature, humidity, motion and illuminance sensors deployed to consumers. All the data used for analysis is anonymized, without means that the real world users could be identified.

As described in the DoA, there are three types of consumers that enCOMPASS project analyzes: households (residential users), public buildings and schools. This platform release deals with data reported by households and public buildings.

The first release (R1) implements the functional specification produced by Task 2.3 and architectural design produced by Task 6.2 that will be enriched and refined in the following releases R2 in M24 and R3 in M36.

Overall, the enCOMPASS platform is composed by software components working together, orchestrated by the system back-end and services. The exposure of its functionality toward the end-users is achieved through the Awareness Applications, both web and mobile, offering user friendly and actionable graphic interfaces for integrating a socio-technical approach to behavioral change for energy saving. The energy consumption data is made available and understandable to the end-users by integrating the visualization of energy data collected from smart sensors, user-generated information and context-aware collaborative recommendations for energy saving, intelligent control and adaptive gamified incentives.

Also, this platform release delivers the first implementation of the first user behavior modeler and recommender produced in Task 4.2

The deliverable itself is made by the source code of the software components making the platform and the documentation for using it.

This document is an accompanying text providing instructions for the installation and running of the enCOMPASS platform.

For a better understanding of this document of and the software deliverable, the following deliverables can be consulted:

- *D6.1 Delivery management plan and testing specification* : contains a description of the software development process; code management tools; documentation to set up the development and testing environments.
- *D6.2 Platform architecture and design*: Describes the architectural design of the enCOMPASS platform: the information and data models, platform components, services, and applications, communication protocols. Integration model.

1 enCOMPASS platform – Initial Prototype

1.1 State of the development process

Figure 1 depicts the UML component representation of the enCOMPASS platform.

The main components of the integrated platform are:

- enCOMPASS platform database;
- Smart Meter and Sensor Data Management (data ingestion component);
- Disaggregation Engine;
- Inference Engine;
- Gamification Engine;
- Efficiency Exploration Console for Utility;
- Recommendation Engine;
- Notification Engine;
- Service Integration and Semaphorization (orchestration component);
- Awareness Applications;
- Efficiency Exploration Console for Building Managers;
- Digital Game Extension.

In the representation of the current state of development of the software components of the enCOMPASS platform, the states of the components have the following meaning:

- Green – completed.
- Yellow – in progress.

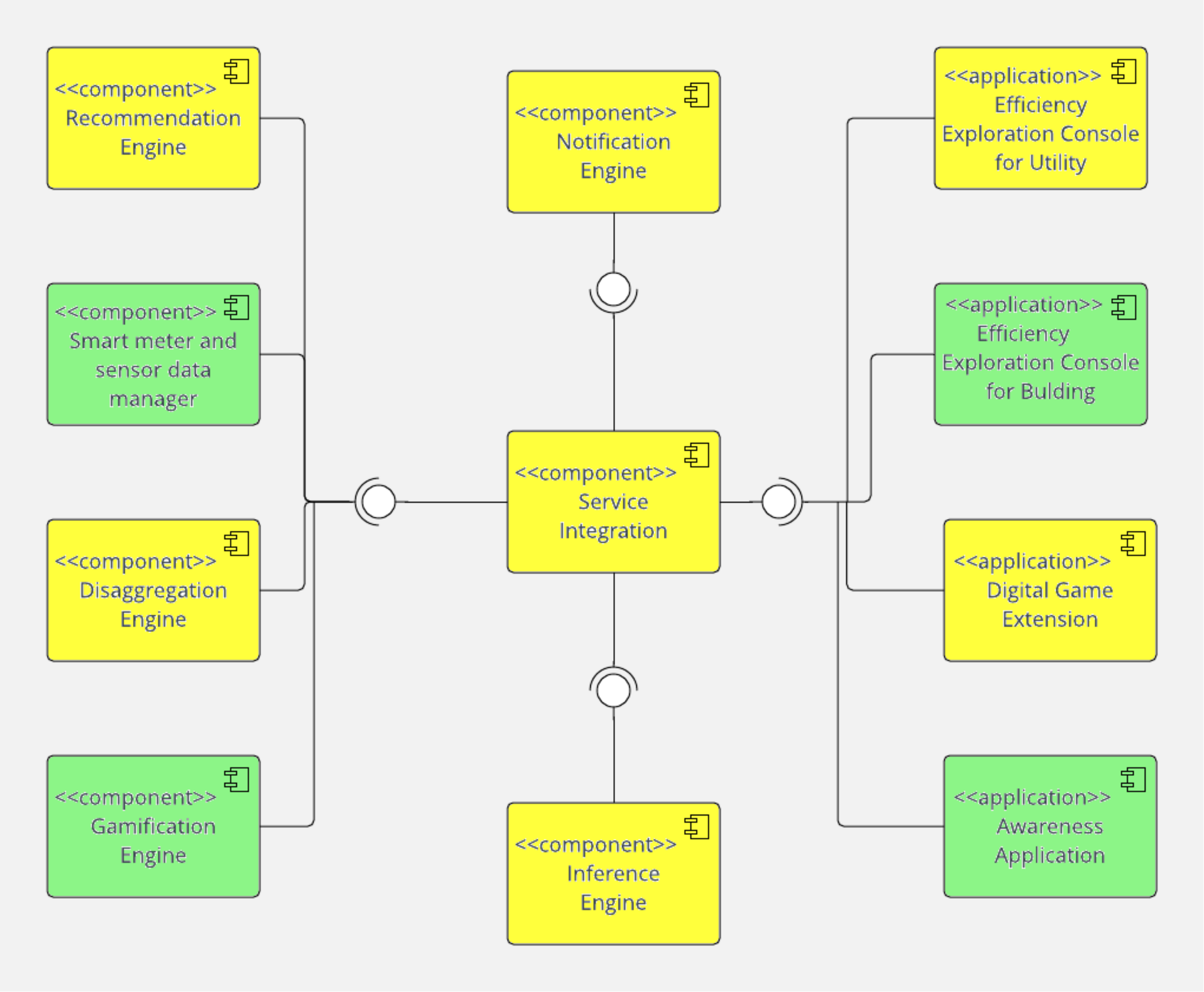


Figure 1 – Overview of the main components of the enCOMPASS architecture and their current state of development.

2 INSTALLATION GUIDE

The software sources referred in the following sections are available as public projects on the enCOMPASS project account on the project development Git server. They can be downloaded and installed from the following addresses:

ssh://18.184.32.200:22/var/git/ AA_android_app.git

ssh://18.184.32.200:22/var/git/ AA_ios_app.git

ssh://18.184.32.200:22/var/git/ encompass-inference-engine.git

ssh://18.184.32.200:22/var/git/ gamification_engine.git

ssh://18.184.32.200:22/var/git/ GamificationEngine_executable.git

ssh://18.184.32.200:22/var/git/encompass-service-integration.git

ssh://18.184.32.200:22/var/git/ encompass-database.git

ssh://18.184.32.200:22/var/git/ gamification_engine_db.git

2.1 ENCOMPASS DATABASE

The **enCOMPASS Database** is the central repository of the information that is either common to all the enCOMPASS components or supports the coordination and exchange of messages among them. Not all the data of enCOMPASS will reside in the enCOMPASS database; for example, commercial data about the energy consumers maintained by the Energy Utilities are stored in the proprietary systems of the respective companies.

The database server for the enCOMPASS platform database is a MySql 5.5+.

The script for creating the database structure is available as Appendix A of the present document.

The script file for creating and initializing the enCOMPASS platform database with a testing data set is available on the Git project account at:

ssh://18.184.32.200:22/var/git/encompass-database.git

2.2 SMART METER AND SENSOR DATA MANAGEMENT OR DATA INGESTION COMPONENT

The **Smart Meter and Sensor Data Manager** (SMSDM) components deals with the acquisition of data streams from smart meters and with their consolidation within the enCOMPASS database. It implements the data privacy and security policy of the utility company and ensures that only admissible (e.g., aggregated, anonymized) data is stored in the platform database.

For accomplishing its task of provisioning data to enCOMPASS platform, SMSDM component process smart meter and sensor readings, uploaded as CSV files by the Energy Utility through SFTP and ingests these data into the platform database. The data ingestion may occur periodically e.g. hourly, daily, weekly, monthly or at any moment the CSV files containing readings are available.
The SMSDM component:

- Retrieves user consumption data from smart meters for past time intervals;
- Retrieves data from sensors (presence, luminosity, temperature, humidity, etc.) for past time intervals.

It is implemented using Apache Camel routes and parallel processing technologies whose main advantage is obtaining scalability when processing increasing amounts of data by just adding and registering new hardware without making any software changes.

This component implements the ETL (Extract, Transform, Load) process with no assumption of the utility of the data, so it can be reused in other data integration projects.

2.2.1 Requirements

- OS: Cent OS 7.0
- Java 8+: https://home.java.net/
- MySQL 5.1+: https://www.mysql.com/
- Apache Maven 3.5+: https://maven.apache.org/download.cgi
- SpringBoot: https://projects.spring.io/spring-boot/

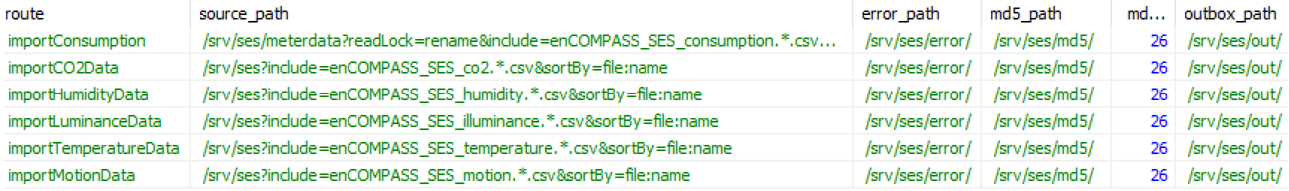
2.2.2 Installation of the SMSDM component

For installing and testing the SMSDMC component, the following steps have to be performed:

1. **Route configuration**

The SMSDM (Data Ingestion) component is a SpringBoot project containing the Apache Camel libraries – an open source Java based framework focusing on data and service integration following a rule-based routing and mediation engine. The first requirement is the configuration of the routes the SMSDM component uses to process the consumption and sensor data files according to the local settings. The route configuration is performed by populating the table **route_config** from the central database of the enCOMPASS platform, as in the sample from Table 1: Populating **route_config** table

*Table 1: Populating **route_config** table*



route	source_path	error_path	md5_path	md...	outbox_path
importConsumption	/srv/ses/meterdata?readLock=rename&include=enCOMPASS_SES_consumption.*.csv...	/srv/ses/error/	/srv/ses/md5/	26	/srv/ses/out/
importCO2Data	/srv/ses?include=enCOMPASS_SES_co2.*.csv&sortBy=file:name	/srv/ses/error/	/srv/ses/md5/	26	/srv/ses/out/
importHumidityData	/srv/ses?include=enCOMPASS_SES_humidity.*.csv&sortBy=file:name	/srv/ses/error/	/srv/ses/md5/	26	/srv/ses/out/
importLuminanceData	/srv/ses?include=enCOMPASS_SES_illuminance.*.csv&sortBy=file:name	/srv/ses/error/	/srv/ses/md5/	26	/srv/ses/out/
importTemperatureData	/srv/ses?include=enCOMPASS_SES_temperature.*.csv&sortBy=file:name	/srv/ses/error/	/srv/ses/md5/	26	/srv/ses/out/
importMotionData	/srv/ses?include=enCOMPASS_SES_motion.*.csv&sortBy=file:name	/srv/ses/error/	/srv/ses/md5/	26	/srv/ses/out/

2. **Project deployment**
   2.1 Make a **git clone** of the master service project from
   ssh://18.184.32.200:22/var/git/encompass-service-integration.git
   2.2 Open a **Command Prompt** and go to the **encompass-data-ingestion** subfolder of the project
   2.3 Configure the **application.properties** file by ensuring the CAMEL PROPERTIES and ROUTE PROPERTIES are set according to local configuration, as in the following configuration sample:

```
#---------------CAMEL PROPERTIES------------------------------
camel.routes.import.consumption=importConsumption  ;energy consumption data ingesting route
camel.routes.import.co2=importCO2Data            ;CO2 sensor data ingesting route
camel.routes.import.humidity=importHumidityData    ;humidity sensor data ingesting route
camel.routes.import.luminance=importLuminanceData  ;illuminance sensor data ingesting route
camel.routes.import.temperature=importTemperatureData ;temperature sensor data ingesting route
camel.routes.import.motion=importMotionData       ;occupancy sensor data ingesting route
camel.routes.import.consumption.warning.null.percentage=10 ;percentage of reading lines having null values when to
trigger an alert for too many nulls
```

```
#---------------/CAMEL PROPERTIES-----------------------------

#--------------MySQL DATABASE PROPERTIES----------------------
spring.datasource.driver-class-name=com.mysql.jdbc.Driver ;MySQL DB driver
spring.datasource.url=jdbc:mysql://18.184.32.200:3306/encompass_model?useUnicode=yes&characterEncoding=UTF
-8&useSSL=false ;MySQL connection URL
spring.datasource.username=puser       ;MySQL username
spring.datasource.password=sj5$mfD&k37z1c;MySQL password
spring.datasource.test-while-idle=true   ;perform ping to avoid disconnection
spring.datasource.test-on-borrow=true    ;perform
spring.datasource.tomcat.max-active=10   ;number of tomcat simultaneous active connections
spring.datasource.max-active=10          ;number of simultaneous active connections
spring.datasource.tomcat.max-wait=10000  ;tomcat connection timeout
spring.datasource.max-wait=10000         ;connection timeout
spring.datasource.validation-query=SELECT 1 ;validation query
#--------------/MySQL DATABASE PROPERTIES---------------------


#--------------MAIL PROPERTIES----------------------
spring.mail.host=smtp.gmail.com                 ;email server address
spring.mail.port=465                            ;email server port
spring.mail.protocol=smtps                      ;email server protocol
spring.mail.username=ses.encompass@gmail.com        ;username for email authentication
spring.mail.password=****************              ;password for email authentication
spring.mail.properties.mail.transport.protocol=smtps ;email server protocol for spring
spring.mail.properties.mail.smtps.auth=true        ;email auth parameter for spring
spring.mail.properties.mail.smtps.starttls.enable=true ;email tls parameter for spring
spring.mail.properties.mail.smtps.timeout=2000       ;email timeout parameter for spring
application.email=encompass.no-reply              ;email address for alert originator
application.email.to=admin@encompass-project.eu      ;email address of the enCOMPASS platform administrator
#--------------/MAIL PROPERTIES----------------------
```

2.4 Launch the command **mvn clean install** to build the project distribution

2.5 Deploy the project on the server

2.6 Launch the server

2.7 In the transfer path specified in the **route_config.source_path** upload CSV files containing consumption and sensor data. The SMSDM component automatically starts processing the file according to the ingestion rules.

2.8 Check the outcome of the SMSDM component in the following tables from the central enCOMPASS database: **semaphore_log, meter_consumption**, **indoor_conditions_humidity**, **indoor_conditions_temperature**, **indoor_conditions_luminanace**, **indoor_conditions_occupancy**.

2.9 Check the processed files in the paths specified in the **route_config.outbox_path** column, if the data ingestion has been successfully performed or respectively in the **route_config.error_path** column if the data ingestion failed.

2.10 Check the email account specified by **application.email.to** configuration variable to consult the success or fail alert message of the data ingesting processing. This alert message is meant for the platform administrator that is in charge with maintain the data ingestion.

2.2.3 Accesso to the source code of the SMSDM component

The NE component source code can be downloaded from the following repository:

ssh://18.184.32.200:22/var/git/encompass-service-integration.git

using the standard Git credentials, then importing the **encompass-data-ingestion** project.

2.3 Disaggregation Engine

This component is not described in this deliverable as according to the Release Plan presented in *D6.1 Delivery management plan and testing specification*; its features are part of the platform second release.

2.4 Inference Engine

The inference engine (IE) is composed of the occupancy and activity inference tool, of the user comfort levels inference tools, and of the scheduler of comfort feedback notifications.

2.4.1 Occupancy and Activity Inference Tool

The Occupancy and Activity Inference Tool is responsible for estimating the number of occupants (or presence/ absence) and the activity performed in a building (e.g. cooking, resting, sleeping, etc.).
The engine takes as input the raw data from the sensors, i.e. the energy consumption and the indoor environmental conditions, and the disaggregated information (i.e. more detailed information regarding the energy consumption in a building).
The engine exploits the data regarding the energy consumption from buildings and devices and fulfills a training process on them. As a result, the engine can infer whether a space is occupied or not, and which activity is performed at a given moment of time.

2.4.2 User Comfort Levels Inference Tool

The Comfort Levels Inference Engine is responsible for estimating the subjective comfort level of the occupants in a building, both thermal and visual. The engine takes as input an interval of dates, ranging from the starting date (year, month and day) until the ending date (year, month and day) and computes the comfort level for all of the users during each of the days in the interval.

For each user, the thermal comfort estimation requires the knowledge of temperature and humidity of a room, as well as the activity and the clothing of the user at a specific moment. The visual comfort estimation requires the luminance level only, during the same moment of time. If a user has given their feedback regarding their desired thermal and visual comfort level, then this feedback is taken into consideration, and contributes to the accuracy of the comfort level estimation.

2.4.3 Scheduler of comfort feedback notifications

The Scheduler of comfort feedback notifications is a tool that regulates when notifications will be sent to users, so that the users respond by sending information about their desired comfort feedback level. If there are notifications that have not been answered by a user (i.e. the user has not given his comfort feedback) until the moment of the execution of the Scheduler, then the dates of these notifications to be delivered are transferred in the current week. Otherwise, a new notification is posted for each user in the current week, which will request comfort feedback from the user.

2.4.4 Requirements

1. MySQL DBMS 5.6 or newer (link)
2. Java 1.8.0 or later (link)
3. Apache Maven 3.5.3 or newer
4. Spring Boot

2.4.5 Installation and execution

1. Download the project from ssh://18.184.32.200:22/var/git/encompass-inference-engine.git and store it in a folder of your choice.
2. In the files that regard the pure components (in Scheduler.java, for example) change the URL of the database and its access credentials (username and password) at your own will.
3. Open a terminal in your computer, and change to the folder where you have stored the project.
4. Build the project by writing `./mvnw clean package`
5. Run the Project by writing `./mvnw spring-boot:run`, which starts Tomcat on port 8083 and the Spring Boot Application.

2.4.6 Testing

This project contains a POST web service, which can be called with any sample JSON as input (even no input at all is accepted). Provided that you have run the project, thus starting Tomcat server on port 8083, the URL you have to use is http://yourIP:8083/ie/component/external/start, where yourIP is the IP address of the machine you will be running this project. When you call this service, the IE components (such as the Scheduler) will be executed.

The IE component source code can be downloaded from the following repository:

ssh://18.184.32.200:22/var/git/encompass-inference-engine.git

using the standard Git credentials.

2.5 Gamification Engine

The **Gamification Engine** is a component that listens to the actions of the users and transforms them into a variety of rewards, for improving activity and participation.

The definition of points, achievements and rewards allow energy utility companies to challenge their users with energy saving goals to be achieved each month to obtain individual energy consumption and data provisioning. The **Gamification Engine** is also used as a mean for raising energy consumption awareness by promoting sustainable behavior.

The Gamification Engine has been developed using Webratio, a platform for the development of web and mobile applications based on the OMG standard IFML.

2.5.1 Requirements

- MySQL DBMS 5.6 (link)
- Java 1.8.0 or later (link)
- Apache Tomcat 6.x or later (link).

2.5.2 Installation

1. **DB Installation.** Create a new database “community” and import the sql file community.sql from ssh:// 18.184.32.200:22/var/git/gamification_engine_db.git
2. **Application installation on Tomcat**.
   - Unzip the ssh:// 18.184.32.200:22/var/git/gamificationEngine_executable.git file and copy the community folder into the webapps folder of your Tomcat installation (henceforth tomcat_webapps);

- Change configuration path of the invoked web services tomcat_webapps/community/WEB-INF/classes/Webratio.properties

```
var services_host_url = "http://18.184.32.200:8083/";
var local_host_url = "http://localhost:8080/"; tomcat address

services_host_url=http://18.184.32.200:8083/si  address of the invoked services
my_host_url=http://localhost tomcat address
my_host_port=8080 tomcat port for basic
my_host_internal_url=http://localhost tomcat address for internal invocations, typically "localhost"
my_host_name=community name of the application
my_host_port_ssl=8443
my_host_internal_port=8080

reward_shipment=false whether rewards are shipped or collected in place
reward_score_decrease=false whether a reward claim causes a score decrease
```

- Change the database configuration (url, username and password), updating the "dbx.hibernate.cfg.xml" file in the tomcat_webapps/community/WEB-INF/classes of the applications:

```
<property name="connection.url">
    jdbc:mysql://localhost:<your_port>/community
</property>
<property name="connection.username">
    <your_user>
</property>
<property name="connection.password">
    <your_password>
</property>
```

- Grant R/W permission to Tomcat to the entire folder **community.** For UNIX:

```
sudo chown -R tomcat7 community/

sudo chgrp tomcat7 community/
```

- Start Tomcat

2.5.3 Testing

The applications can be accessed using the following url:

- http://localhost:<your_port>/community (customer frontend). The portal can be accessed using the following credentials:
- Advanced profile

  username: Chiara

  password: password

- https://localhost:< your_port >/community/admin (admin frontend). The portal can be accessed using the following credentials:

username: admin

password: password

2.5.4 Source code

The Webratio projects can be downloaded from:

ssh://18.184.32.200:22/var/git/gamification_engine.git

and can be imported into the Webratio IDE using the projects:

- Authentication,
- BootstrapStyleRarolab,
- Gamification,
- GamificationBackEndStyle,
- GamificationCustomRarolab,
- GamificationFrontEndStyle,
- NotificationMessageSample.

2.6 Energy Efficiency Console Platform for Utility

This component performs the monitoring of consumption behaviour and household characteristics. It is included in the Gamification Engine project as a user access role. The installation of this component is already performed when installing the Gamification Engine component.

2.7 Recommendation Engine

This component produces action recommendations based on disaggregated consumption and sensor data. Proposed recommendations will be sent to Central database to be further sent to End User by Notification Engine.
The Recommendation Engine ingests the following data available in the Platform:

- consumption and sensor data;
- user profile data;
- building information;
- user actions on the Platform including user responses to Recommendation Engine suggestions.

Recommendation Engine backed is not an integral part of the encompass platform, it is a SaaS like service. Its installation and monitoring are provided by Gravity on their own server.

The Recommendation backend in high level is an API-less service which takes input in the form of an uploaded directory of database exports from the encompass platforms, calculates recommendations for the next day on it, and then sends next day's recommendations to the encompass platform in the form of an API call of the postMessageInstanceList.

To prevent recalculation and resending the data (data import is done in every case), internally the Recommendation Backed restricts reruns of these tasks to once a day.

Because there is no real trigger mechanism (because there is no API), the Recommendation Engine backend is running as a cronjob on the server and will check the existence of a [date].go directory in the SFTP directory, and if it exists, it will import the data, calculate, and send the recommendation.

The state of the import can be checked externally by connecting to the SFTP server (credentials has been provided in a separate encrypted archive) and checking the state of the data import (not the whole recommendation calculation process). It has four states:

- [date].go: the Recommendation Engine not yet started to process this directory
- [date].processing: the Recommendation Engine is importing the data
- [date].failed: the Recommendation Engine is failed to import the data
- [date].finished: the Recommendation Engine is finished with the data import

Because of the orchestration, the cronjob that is running the Recommendation Engine is running every minute, which will prevent large delay between data export from the enCompass platform, and the data import in the Recommendation Engine.

Also, to make orchestration work in every case, the Recommendation Engine backend is implemented in a way that even when the calculation did not run (because of the calculation once per day restriction), the call to the API which start the next task in the orchestration will be called.

Configuration of the Recommendation Engine backend is also internal, any changes to it (like changing server address, cron timers, etc.) has be done by Gravity operation team.

2.8 Notification Engine

This component is responsible for sending PUSH notifications to End User Client application, like Awareness Application. It reads messages to be sent from the Central Database and queues them for sending. Any Platform component that needs to send a message needs to store the message in a message table specifying the destination application (endpoint), message content and the schedule.

2.8.1 Requirements

- OS: Cent OS 7.0
- Java 8+: https://home.java.net/
- MySQL 5.1+: https://www.mysql.com/
- Apache Maven 3.5+: https://maven.apache.org/download.cgi
- SpringBoot: https://projects.spring.io/spring-boot/
- ActiveMQ 5.15+: http://activemq.apache.org/download.html

2.8.2 Installation

For installing the NE component, the following steps have to be performed:

1. **Project deployment**

   1.1 Make a **git clone** of the master service project from **ssh://18.184.32.200:22/var/git/encompass-service-integration.git**
   1.2 Open a **Command Prompt** and go to the **encompass-component-services** subfolder of the project
   1.3 Configure the **application.properties** file according to local configuration, as in the following configuration sample:

```
#---------------ACTIVEMQ--------------------------------------
spring.activemq.in-memory=false                ;parameter
spring.activemq.pool.enabled=false             ;parameter
activemq.broker.url=tcp://localhost:61616        ;message broker connection  protocol
activemq.queue=queue://                        ;internal storage structure
activemq.queue.low=notification.lowpriority.queue   ;low priority queue id
activemq.queue.high=notification.highpriority.queue
activemq.queue.onthefly=notification.onthefly.queue
#---------------/ACTIVEMQ------------------------------------

#--------------MySQL DATABASE PROPERTIES----------------------
spring.datasource.driver-class-name=com.mysql.jdbc.Driver ;MySQL DB driver
spring.datasource.url=jdbc:mysql://18.184.32.200:3306/encompass_model?useUnicode=yes&characterEncoding=UTF
-8&useSSL=false ;MySQL connection URL
spring.datasource.username=puser       ;MySQL username
spring.datasource.password=sj5$mfD&k37z1c;MySQL password
spring.datasource.test-while-idle=true   ;perform ping to avoid disconnection
spring.datasource.test-on-borrow=true    ;perform
spring.datasource.tomcat.max-active=10   ;number of tomcat simultaneous active connections
spring.datasource.max-active=10        ;number of simultaneous active connections
spring.datasource.tomcat.max-wait=10000  ;tomcat connection timeout
spring.datasource.max-wait=10000       ;connection timeout
spring.datasource.validation-query=SELECT 1 ;validation query
#--------------/MySQL DATABASE PROPERTIES---------------------


#---------------FIREBASE PROPERTIES---------------------------
firebase.api.key=AAAA1Nqhg9E:APA91bHSIFc7WoQKNkgCF1AhvTd_nncEO3M3GIQEtXRm-
nVP7YRqMcPwjTfrcLvk72m0kvF5DTw4JYuAmTKbXuvq_ozx7OtAizovlUOH-Oyeo8yoRjPoVNyjzTdCspWQeTNzEkre-
Ts3   ;Firebase project API Key
firebase.api.project-id=914201084881  ;Firebase project ID
firebase.endpoint.operation.group=https://android.googleapis.com/gcm/notification  ;endpoint for creating Firebase
group
firebase.endpoint.operation.message=https://fcm.googleapis.com/fcm/send ;endpoint for sending notification
firebase.prefix.group=encompass-     ;Firebase group prefix
#--------------/FIREBASE PROPERTIES--------------------------
```

1.4 Launch the command **mvn clean install** to build the project distribution
1.5 Deploy the project distribution to the server
1.6 Launch the server

2.8.3 Testing

After successfully deploying the NE component, the ActiveMQ console is available for testing at the URL http://<IP>:8161/ . For the development and testing server, the ActiveMQ console is available at http://18.184.32.200:8161/admin/

2.8.4 Source code

The NE component source code can be downloaded from the following repository:

ssh://18.184.32.200:22/var/git/encompass-service-integration.git

using the standard Git credentials, then importing the **encompass-component-services** project.

2.9 Service Integration and Component Orchestration (Semaphorization)

This component manages all interaction between Platform components and between Platform component and Central Database. All interactions will be done only through Web Services that will be enlisted in this component.

2.9.1 Requirements

- OS: Cent OS 7.0
- Java 8+: https://home.java.net/
- MySQL 5.1+: https://www.mysql.com/
- Apache Maven 3.5+: https://maven.apache.org/download.cgi
- SpringBoot: https://projects.spring.io/spring-boot/

2.9.2 Installation

For installing the Service Integration and Semaphoring component, the following steps have to be performed:

1. **Project deployment**

   1.1 Make a **git clone** of the master service project from **ssh://18.184.32.200:22/var/git/encompass-service-integration.git**

   1.2 Open a **Command Prompt** and go to the **encompass-component-services** subfolder of the project

   1.3 Configure the **application.properties** file according to local configuration, as in the following configuration sample:

```
#---------------SEMAPHORE--------------------------------------
semaphore.component.internal.url=http://18.184.32.200:8081/%s/component/start ;endpoint for starting the next component within the component orchestration
semaphore.component.ie.url=http://18.184.32.200:8083/ie/component/external/start ;endpoint for starting the Inference Engine component within the component orchestration
semaphore.post_component_subprocess_result_url=http://18.184.32.200:8081/si/semaphore/subprocess/result ;endpoint for saving a subprocess result within the component orchestration
semaphore.post_component_processing_finalization_url=http://18.184.32.200:8081/si/semaphore/result ;endpoint for saving a process finalization and return the control to semaphore within the component orchestration

#---------------/SEMAPHORE--------------------------------------

#--------------MySQL DATABASE PROPERTIES----------------------
spring.datasource.driver-class-name=com.mysql.jdbc.Driver ;MySQL DB driver
spring.datasource.url=jdbc:mysql://18.184.32.200:3306/encompass_model?useUnicode=yes&characterEncoding=UTF-8&useSSL=false ;MySQL connection URL
spring.datasource.username=puser       ;MySQL username
spring.datasource.password=sj5$mfD&k37z1c;MySQL password
spring.datasource.test-while-idle=true   ;perform ping to avoid disconnection
spring.datasource.test-on-borrow=true    ;perform
spring.datasource.tomcat.max-active=10   ;number of tomcat simultaneous active connections
spring.datasource.max-active=10         ;number of simultaneous active connections
spring.datasource.tomcat.max-wait=10000  ;tomcat connection timeout
spring.datasource.max-wait=10000        ;connection timeout
spring.datasource.validation-query=SELECT 1 ;validation query
#--------------/MySQL DATABASE PROPERTIES---------------------

#---------------SWAGGER PROPERTIES----------------------------
swagger.info.title=ENCOMPASS COMPONENTS API
swagger.info.description=ENCOMPASS COMPONENTS API
swagger.info.version=v0.1
swagger.info.terms=Terms
swagger.info.contact.name=enCompass
swagger.info.contact.url=http://www.encompass-project.eu
swagger.info.contact.mail=contact@encompass-project.eu
swagger.info.license=enCompass License
swagger.info.license.url=http://www.encompass-project.eu
#--------------/SWAGGER PROPERTIES---------------------------
```

1.4 Launch the command **mvn clean install** to build the project distribution
1.5 Deploy the project distribution to the server
1.6 Launch the server

2.9.3 Testing

After successful deploying the Service Integration and Semaphoring component, the Swagger console is available at the URL http://<IP>: 8081/swagger-ui.html . For the development and testing server, the Swagger console is available at http://18.184.32.200:8081/swagger-ui.html

2.9.4 Source code

The Service Integration and Semaphoring component source code can be downloaded from the following repository:

ssh://18.184.32.200:22/var/git/encompass-service-integration.git

using the standard Git credentials, then importing the **encompass-component-services** project.

2.10 Awareness Applications

The enCOMPASS platform enables its users to track their energy consumption and obtain useful information that helps them to improve their energy consumption efficiency while keeping it comfort level and having fun with gamified activities.

Such interaction with the user is done through a web-based application and a mobile client (available for android and iOS). These components provide to End User the following main functionalities:

- Consumption data tracking;
- Energy reduction goal setting and tracking
- Tip and personalized recommendation reading
- Achievement tracking and leaderboard visualization
- Energy saving impact
- Comfort visualization and feedback
- Event Notification Delivery.

The Awareness Application interacts directly with the Gamification Engine to provide the Gamification functionalities, and with the Service Integration component retrieve the consumption, comfort, and awareness information (tips and recommendations).

The mobile clients code can de find in the following repositories:

Android: ssh://18.184.32.200:22/var/git/encompass-androidApp.git

iOS: ssh://18.184.32.200:22/var/git/encompass-ios.git

2.10.1 Requirements

Android:

- Android 3.0 or later
- Java JDK 1.7 or later
- GIT 2.0 or later

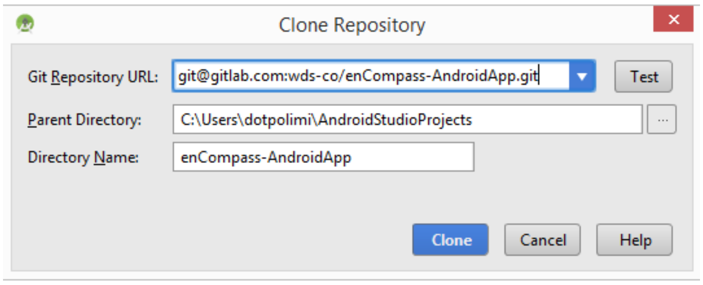
iOS:

- macOS 10.13.2 or later
- XCode 9.1
- GIT 2.0 or later

2.10.2 Installation of the Android project

In order to get the application code into Android Studio, create a new project, from the Menu File, select New – Project from Version Control – Git.

On the pop-up window, provide the above URL as the Git Repository URL, and select a path to save the project as in Figure 2. Click clone and provide the require user and password when prompt.



Git Repository URL: git@gitlab.com:wds-co/enCompass-AndroidApp.git

Parent Directory: C:\Users\dotpolimi\AndroidStudioProjects

Directory Name: enCompass-AndroidApp

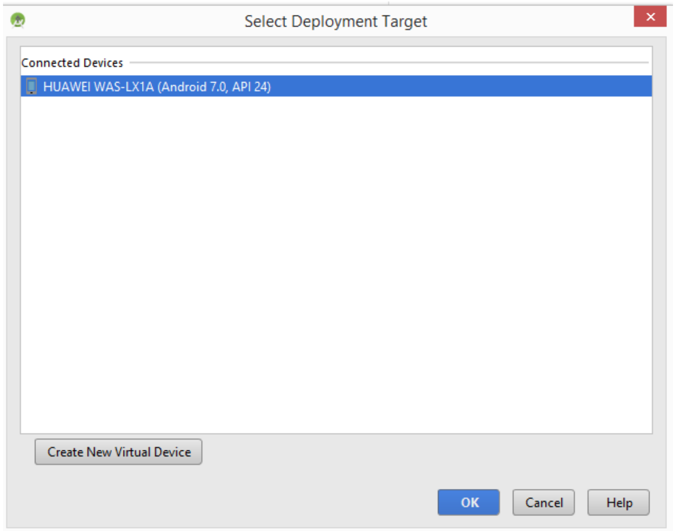
Figure 2 – Clonning the Awareness Application repository

Once the cloning process is finish, the build process will execute automatically, in case it does not it can be start manually by going to the menu Build –> rebuild.

2.10.3 Testing the Android app

Now the application can be tested in 2 ways:

1. Running the app on a connected device, as seen in Figure 3, or on a device Emulator: Go the menu Run –> run, a window will display a list of available devices (physical or virtual), select one and click ok. Android Studio will compile, pack and install the application on the selected device, the application will execute once the installation process is finish.



Select Deployment Target

Connected Devices

HUAWEI WAS-LX1A (Android 7.0, API 24)

Figure 3 – Runnning the Awareness Application on a connected device

2. Generate an APK and install manually on a device: Android studio can generate and Android Package for the application, to do so go to the menu Build –> Build APK(s). The apk file will be created automatically on the /app/build/outputs/apk folder. Copy the file to the testing device, and click on it, a pop-up menu will ask to trust the application and allow installation from unknow sources, click on accept on the device. The application will be installed and can be executed.

Notice that the application was built to support android API level 25 (Android Nougat 7.1) and minimum supported API level is 21 (Android Lollipop 5.0) which provides a wide range of supported devices, this information can be found on the build file (/app/build.gradle) but changing it would result in compilation errors.

Additionally the application uses Firebase Cloud Messaging Services, which are automatically configure by android studio base on the content of the file google-services.json, located on the /app folder, Firebase services will be detail in the following sections.

2.10.4 Android Client Configuration

In order to configure the Android client to work with different backend servers the following files need to be changed:

String.xml:

The file /app/src/main/res/values/String.xml contains the list of available providers, in order to add and remove. In the property "provider_array":

```
<string-array name="provider_array">
    <item>Service provider</item>
    <item>SES</item>
    <item>SHF</item>
    <item>WTV</item>
</string-array>
```

Config.properties:
This file contains the configuration of the servers corresponding to the previous list, therefore they should be aligned. The file is located in: /app/src/main/assets/config.properties.

The properties represent the host server of the web application, and they are divided as protocol, host and port. Each property has a prefix corresponding with the provider, as follows:

```
ses.protocol=http://
ses.host=18.184.32.200
ses.port=:8080
```

2.10.5 App Store Deployment

Deployment to Google Play Store for beta testing and distribution requires to register as Developer at the play store by paying a licensing fee. After registration the development console provides with certificates that should be used to sign the APK file to be distributed by google. Full details of this process can be found at google developers and android developers.

2.10.6 Installation of the iOS Project

To get the code of the iOS project, it is necessary to clone the git repository on your local environment. Create a folder, and go to that location on a terminal, execute the following command:

```
git clone ssh: ://18.184.32.200:22/var/git/encompass-ios.git
```

Once the process is finish open Xcode 9, go to the menu File –> Open, navigate to the folder where the repository was copied, open the file enCOMPASS.xcworkspace and click open. Note that the project uses a series of Pods and libraries, while clicking the file .xcodeproj will open the app code, it will not load these dependencies.

2.10.7 Testing the iOS app

The application can be tested in 2 ways:

1. Installing the app in a device emulator: In Xcode go to the menu Product –> Destination iOS Simulators, and select the device to emulate, Xcode will download the device emulation files, once finish click on the menu Product –> Run.
2. Installing the app in a physical device: Connect a device via USB to the computer, go to the menu Product –> Destination –> Device, the device will be listed in the menu, select it and go to menu Product –> Run.

Consider that the app was built to work with iOS 11 and later, this can be see in the main project configuration view, changing the target deployment will generate compilation errors and warnings.

2.10.8 iOS Configuration

In order to configure the iOS client to work with other backend servers the data.plist file need to be changed. The file is located on the path /EnCOMPASS/encompass/src/data.plist.

The property providersArray, contains the list of the service providers that are shown in the login view. In order to add or remove providers the property has to be change.

```
<key>prodiverList</key>
<array>
      <string>SES</string>
      <string>SHF</string>
      <string>WVT</string>
</array>
```

Then separate properties for the servers, corresponding to each provider, can be found in the same file, with the following syntax:

```
<key>ses.protocol</key>
<string>http://</string>
<key>ses.host</key>
<string>18.184.32.200</string>
<key>ses.port</key>
<string>:8080</string>
```

The properties have been separated in 3 parts to allow secure or regular http protocol, and different ports that might be use by different application servers. Both the properties and the provider array above should be in line.

2.10.9 Development and App Store Deployment

In order to download the tools and be able to test the application on a device, registration on the Apple developer site is required: https://developer.apple.com/.

Deployment to apple TestFlight for beta testing and apple store for distribution requires to registration to the apple development program at the developer website by paying a licensing fee. After registration the console provides with certificates that should be used to sign the application file to be distributed by apple. Full details of this process can be found at program website and enrollment procedure site.

2.10.10 Firebase Cloud Messaging

As mention before, the gamification engine uses google Firebase services to deliver messages that are present by the clients as notifications. In order to create and setup a Firebase project, go to the Firebase Console with a google user, create a new project, and provide a name.

Go to the project settings and create 2 applications one for Android and one for iOS:

- Android: The android package name will be requested, make sure to provide the app id of the android application: “com.eu.encompass”
- iOS: The App bundle id will be requested, provide the id on the iOS project: “it.polimi.encompass”

If the above properties are not aligned with the corresponding application ids the messages will not be deliver to the clients.

Configuration

Once the project is configured, firebase will generate a json configuration file for each app, it should be included with the code of each app to authenticate the client with the backend servers.

For android go the project settings in the firebase console, and select the android application, and download the google-service.json file. Copy the file into the android project on the /app folder.

For iOS repeat the process, go to the firebase console and select the iOS application, download the googleServices-info.plist file, finally add the file to the iOS project. Select enCompass project on the file navigation of Xcode, on the menu File –> Add Files to “enCOMPASS” and select the google services file.

In order to fully configure the iOS app for Firebase, the APN (Apple Push Notification) certificate need to be created on the apple developer console on the section Certificates, Identifiers & Profiles, 2 certificates need to be created 1 for development and testing and 1 for distribution.

Once created the certificates need to be uploaded to firebase console, on the project overview –> settings –> Cloud Messaging –> iOS application –> APNs Certificate.

Fully detail process of how to create the certificates and upload them to firebase can be found in the following links: Configuring APNs with FCM

2.11 Energy Efficiency Console Platform for Buildings Managers

The **Energy Efficiency Console Platform for Buildings Managers** is a platform where the user can see information (e.g. energy consumption, indoor environmental, etc.) from the building he/she is owner or manager. He is able to see the current status as well as historical information. Furthermore, the platform supports multiple dashboards, which are fully parameterized, where the user can visualize different data depending on his/her interests.

The platform has been based on Java for the backend, while the frontend has been based on Angular-1 with parts developed at PhP.

The platform is secured and is hosted at https://encompass.pragma-iot.com

The credentials for demo purposes are:

Login: encompass-demo

Password: encomp@$$-d3mo

2.12 Digital Game Extension

This component is represented by the mobile game that bridges the enCOMPASS platform with the board-game. The development of the mobile game has been started and it is planned to be released as part of the Platform Second Prototype (Release 2). The evolution of the component at the current moment (M18) is detailed in deliverable D2.3 Functional System Specifications.

3 APPENDIX A

enCOMPASS DATABASE CREATION SCRIPT

```
-- --------------------------------------------------------
-- Host:                         18.184.32.200
-- Server version:               5.5.58 - MySQL Community Server (GPL)
-- Server OS:                    Linux
-- HeidiSQL Version:             9.5.0.5196
-- --------------------------------------------------------

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET NAMES utf8 */;
/*!50503 SET NAMES utf8mb4 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;

-- Dumping database structure for encompass_model
CREATE DATABASE IF NOT EXISTS `encompass_model` /*!40100 DEFAULT CHARACTER SET utf16 */;
USE `encompass_model`;

-- Dumping structure for table encompass_model.activity_inference
CREATE TABLE IF NOT EXISTS `activity_inference` (
  `oid` int(11) NOT NULL,
  `dwelling_room_oid` int(11) DEFAULT NULL,
  `activity` varchar(45) DEFAULT NULL,
  `timestamp` datetime DEFAULT NULL,
  `timestamp_received` datetime DEFAULT NULL,
  PRIMARY KEY (`oid`),
  KEY `fk_activity_dwelling_room_idx` (`dwelling_room_oid`),
  CONSTRAINT `fk_activity_dwelling_room` FOREIGN KEY (`dwelling_room_oid`) REFERENCES `dwelling_room` (`oid`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- Dumping structure for table encompass_model.baseline
CREATE TABLE IF NOT EXISTS `baseline` (
  `oid` int(11) NOT NULL AUTO_INCREMENT,
  `smart_meter_oid` int(11) NOT NULL,
  `total_consumption` decimal(19,3) DEFAULT NULL,
  `month` date DEFAULT NULL,
  PRIMARY KEY (`oid`),
  KEY `fk_baseline_smart_meter1_idx` (`smart_meter_oid`),
  CONSTRAINT `fk_baseline_smart_meter1` FOREIGN KEY (`smart_meter_oid`) REFERENCES `smart_meter` (`oid`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB AUTO_INCREMENT=516 DEFAULT CHARSET=utf8;


-- Dumping structure for table encompass_model.baseline_import
CREATE TABLE IF NOT EXISTS `baseline_import` (
  `oid` int(11) NOT NULL AUTO_INCREMENT,
  `smart_meter_id` varchar(255) NOT NULL,
  `total_consumption` decimal(19,3) DEFAULT NULL,
  `month` date DEFAULT NULL,
  PRIMARY KEY (`oid`)
) ENGINE=InnoDB AUTO_INCREMENT=303 DEFAULT CHARSET=utf8;


-- Dumping structure for table encompass_model.building
CREATE TABLE IF NOT EXISTS `building` (
  `oid` int(11) NOT NULL AUTO_INCREMENT,
  `district_oid` int(11) DEFAULT NULL,
  `name` varchar(255) DEFAULT NULL,
  `construction_year` year(4) DEFAULT NULL,
  `building_size` decimal(19,2) DEFAULT NULL,
  `address` varchar(255) DEFAULT NULL,
  `renovation_year` year(4) DEFAULT NULL,
  `building_type_oid` int(11) DEFAULT NULL,
  PRIMARY KEY (`oid`),
  KEY `fk_building_district` (`district_oid`),
  KEY `fk_building_type_idx` (`building_type_oid`),
  CONSTRAINT `fk_building_type` FOREIGN KEY (`building_type_oid`) REFERENCES `building_type` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION,
  CONSTRAINT `fk_building_district` FOREIGN KEY (`district_oid`) REFERENCES `district` (`oid`)
) ENGINE=InnoDB AUTO_INCREMENT=184 DEFAULT CHARSET=utf8;
```

```
-- Dumping structure for table encompass_model.building_type
CREATE TABLE IF NOT EXISTS `building_type` (
  `id` int(11) NOT NULL,
  `language` enum('en','it','de','gr') NOT NULL,
  `text` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`id`,`language`)
) ENGINE=InnoDB DEFAULT CHARSET=utf16;


-- Dumping structure for procedure encompass_model.calculate_motivation
DELIMITER //
CREATE DEFINER=`root`@`%` PROCEDURE `calculate_motivation`(IN id INT, money INT, env INT,  fun INT)
BEGIN


        declare motivation int;


        IF  money >= fun AND  money > env AND money >= 3
                THEN SET motivation=1;
        ELSE
                IF  env >= fun AND  env >= money AND env >= 3
                        THEN SET motivation=2;
                ELSE
                        IF  fun > env AND  fun > money  AND fun >= 3
                                THEN SET motivation=3;
                        ELSE
                                SET motivation=2;
                        END IF;
                END IF;
        END IF;


   UPDATE `user` SET `motivation`= motivation , `saving_goal_money` = money, `saving_goal_fun` = fun, `saving_goal_environmental` = env WHERE `oid`=id;
```

```
END//
DELIMITER ;

-- Dumping structure for table encompass_model.complex_device_instance
CREATE TABLE IF NOT EXISTS `complex_device_instance` (
 `oid` int(11) NOT NULL AUTO_INCREMENT,
 `dwelling_room_oid` int(11) DEFAULT NULL,
 `device_type_oid` int(11) DEFAULT NULL,
 `efficiency` varchar(255) DEFAULT NULL,
 `ecomode` bit(1) DEFAULT NULL,
 `timer` bit(1) DEFAULT NULL,
 `nominal_power` decimal(19,3) DEFAULT NULL,
 `remote_control` bit(1) DEFAULT NULL,
 PRIMARY KEY (`oid`),
 KEY `fk_complex_device_instance_dev` (`device_type_oid`),
 KEY `fk_complex_device_instance_hou_idx` (`dwelling_room_oid`),
 CONSTRAINT `fk_complex_device_instance_dev` FOREIGN KEY (`device_type_oid`) REFERENCES `device_type` (`oid`),
 CONSTRAINT `fk_complex_device_instance_hou` FOREIGN KEY (`dwelling_room_oid`) REFERENCES `dwelling_room` (`oid`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- Dumping structure for table encompass_model.cost_estimation
CREATE TABLE IF NOT EXISTS `cost_estimation` (
 `oid` int(11) NOT NULL AUTO_INCREMENT,
 `dwelling_oid` int(11) DEFAULT NULL,
 `date_start` date DEFAULT NULL,
 `date_end` date DEFAULT NULL,
 `unit_eur_charge` decimal(19,3) DEFAULT NULL,
 PRIMARY KEY (`oid`),
 KEY `fk_bill_household` (`dwelling_oid`),
 CONSTRAINT `fk_bill_household` FOREIGN KEY (`dwelling_oid`) REFERENCES `dwelling` (`oid`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- Dumping structure for table encompass_model.device_consumption
CREATE TABLE IF NOT EXISTS `device_consumption` (
 `oid` int(11) NOT NULL AUTO_INCREMENT,
 `complex_device_instance_oid` int(11) DEFAULT NULL,
 `simple_device_instance_oid` int(11) DEFAULT NULL,
 `datetime` datetime DEFAULT NULL,
 `datetime_received` datetime DEFAULT NULL,
 `consumption` decimal(19,3) DEFAULT NULL,
 `is_disaggregated` tinyint(1) DEFAULT NULL,
 PRIMARY KEY (`oid`),
 KEY `fk_simple_cons_idx` (`simple_device_instance_oid`),
 KEY `fk_complex_cons_idx` (`complex_device_instance_oid`),
 CONSTRAINT `fk_complex_cons` FOREIGN KEY (`complex_device_instance_oid`) REFERENCES `complex_device_instance` (`oid`) ON DELETE NO ACTION ON UPDATE NO ACTION,
 CONSTRAINT `fk_simple_cons` FOREIGN KEY (`simple_device_instance_oid`) REFERENCES `simple_device_instance` (`oid`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB DEFAULT CHARSET=utf8;


-- Dumping structure for table encompass_model.device_reading
CREATE TABLE IF NOT EXISTS `device_reading` (
 `oid` int(11) NOT NULL AUTO_INCREMENT,
 `complex_device_instance_oid` int(11) DEFAULT NULL,
 `simple_device_instance_oid` int(11) DEFAULT NULL,
 `datetime` datetime DEFAULT NULL,
 `datetime_received` datetime DEFAULT NULL,
 `reading` decimal(19,3) DEFAULT NULL,
 `is_disaggregated` tinyint(1) DEFAULT NULL,
 PRIMARY KEY (`oid`),
 KEY `fk_device_consumption_complex_device_instance` (`complex_device_instance_oid`),
 KEY `fk_device_consumption_simple_device_instance` (`simple_device_instance_oid`),
 CONSTRAINT `fk_device_consumption_complex_device_instance` FOREIGN KEY (`complex_device_instance_oid`) REFERENCES `complex_device_instance` (`oid`),
 CONSTRAINT `fk_device_consumption_simple_device_instance` FOREIGN KEY (`simple_device_instance_oid`) REFERENCES `simple_device_instance` (`oid`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;


-- Dumping structure for table encompass_model.device_type
CREATE TABLE IF NOT EXISTS `device_type` (
 `oid` int(11) NOT NULL AUTO_INCREMENT,
 `name` varchar(255) DEFAULT NULL,
 `icon` text,
 `type` varchar(255) DEFAULT NULL,
 PRIMARY KEY (`oid`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- Dumping structure for table encompass_model.district
CREATE TABLE IF NOT EXISTS `district` (
 `oid` int(11) NOT NULL AUTO_INCREMENT,
 `name` varchar(255) DEFAULT NULL,
 `country` varchar(255) DEFAULT NULL,
 `city` varchar(255) DEFAULT NULL,
 `zipcode` varchar(45) DEFAULT NULL,
 `timezone` int(11) DEFAULT NULL,
 PRIMARY KEY (`oid`)
) ENGINE=InnoDB AUTO_INCREMENT=46 DEFAULT CHARSET=utf8;

-- Dumping structure for table encompass_model.dwelling
CREATE TABLE IF NOT EXISTS `dwelling` (
 `oid` int(11) NOT NULL AUTO_INCREMENT,
 `building_oid` int(11) NOT NULL,
 `name` varchar(255) DEFAULT NULL,
 `dwelling_size` decimal(19,2) DEFAULT NULL,
 `ownership` bit(1) DEFAULT NULL,
 `public` bit(1) DEFAULT NULL,
 `number_rooms` int(11) DEFAULT NULL,
 `number_adults` int(11) DEFAULT NULL,
 `number_children` int(11) DEFAULT NULL,
 `number_visitors` int(11) DEFAULT NULL,
 `windows_position` varchar(255) DEFAULT NULL,
 PRIMARY KEY (`oid`),
 KEY `fk_dwelling_building1_idx` (`building_oid`),
 CONSTRAINT `fk_dwelling_building1` FOREIGN KEY (`building_oid`) REFERENCES `building` (`oid`) ON DELETE NO ACTION
ON UPDATE NO ACTION
) ENGINE=InnoDB AUTO_INCREMENT=184 DEFAULT CHARSET=utf8;

-- Dumping structure for table encompass_model.dwelling_room
```

```
CREATE TABLE IF NOT EXISTS `dwelling_room` (
  `oid` int(11) NOT NULL AUTO_INCREMENT,
  `dwelling_oid` int(11) NOT NULL,
  `name` varchar(255) DEFAULT NULL,
  `area` decimal(19,3) DEFAULT NULL,
  PRIMARY KEY (`oid`),
  KEY `fk_dwelling_rooms_dwelling1_idx` (`dwelling_oid`),
  CONSTRAINT `fk_dwelling_rooms_dwelling1` FOREIGN KEY (`dwelling_oid`) REFERENCES `dwelling` (`oid`) ON DELETE NO
ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB AUTO_INCREMENT=196 DEFAULT CHARSET=utf8;


-- Dumping structure for table encompass_model.dwelling_type
CREATE TABLE IF NOT EXISTS `dwelling_type` (
  `id` int(11) NOT NULL,
  `language` enum('en','it','de','gr') NOT NULL,
  `text` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`id`,`language`)
) ENGINE=InnoDB DEFAULT CHARSET=utf16;


-- Dumping structure for table encompass_model.generic_message
CREATE TABLE IF NOT EXISTS `generic_message` (
  `oid` int(11) NOT NULL AUTO_INCREMENT,
  `automatic_exec` tinyint(1) DEFAULT NULL,
  `apply_to` varchar(255) DEFAULT NULL,
  `message_type` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`oid`),
  KEY `message_type` (`message_type`)
) ENGINE=InnoDB AUTO_INCREMENT=77 DEFAULT CHARSET=utf8;


-- Dumping structure for view encompass_model.get_tips
-- Creating temporary table to overcome VIEW dependency errors
CREATE TABLE `get_tips` (
	`user_oid` INT(11) NULL,
	`id` INT(11) NOT NULL,
	`name` VARCHAR(45) NULL COLLATE 'utf8_general_ci',
	`header` CHAR(0) NOT NULL COLLATE 'utf8_general_ci',
		`body` VARCHAR(255) NULL COLLATE 'utf8_general_ci',
		`image` VARCHAR(255) NULL COLLATE 'utf8_general_ci',
		`video` VARCHAR(255) NULL COLLATE 'utf8_general_ci',
		`instance` INT(11) NOT NULL,
		`feedback` VARCHAR(255) NULL COLLATE 'utf8_general_ci',
		`static` TINYINT(4) NULL,
		`timestamp_creation` DATETIME NULL
) ENGINE=MyISAM;

-- Dumping structure for view encompass_model.get_tips_with_feedback
-- Creating temporary table to overcome VIEW dependency errors
CREATE TABLE `get_tips_with_feedback` (
		`oid` INT(11) NOT NULL,
		`feedback` VARCHAR(255) NULL COLLATE 'utf8_general_ci',
		`message_instance_oid` INT(11) NOT NULL
) ENGINE=MyISAM;

-- Dumping structure for table encompass_model.group
CREATE TABLE IF NOT EXISTS `group` (
  `oid` int(11) NOT NULL AUTO_INCREMENT,
  `groupname` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`oid`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- Dumping structure for table encompass_model.heating_source_type
CREATE TABLE IF NOT EXISTS `heating_source_type` (
  `id` int(11) NOT NULL,
  `language` enum('en','it','de','gr') NOT NULL,
  `text` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`id`,`language`)
) ENGINE=InnoDB DEFAULT CHARSET=utf16;

-- Dumping structure for table encompass_model.heating_type
CREATE TABLE IF NOT EXISTS `heating_type` (
  `id` int(11) NOT NULL,
  `language` enum('en','it','de','gr') NOT NULL,
  `text` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`id`,`language`)
) ENGINE=InnoDB DEFAULT CHARSET=utf16;


-- Dumping structure for table encompass_model.indoor_conditions_co2
CREATE TABLE IF NOT EXISTS `indoor_conditions_co2` (
  `oid` int(11) NOT NULL AUTO_INCREMENT,
  `dwelling_room_oid` int(11) DEFAULT NULL,
  `value` double DEFAULT NULL,
  `datetime` datetime DEFAULT NULL,
  `datetime_received` datetime DEFAULT NULL,
  PRIMARY KEY (`oid`),
  KEY `fk_environmental_room_idx` (`dwelling_room_oid`),
  CONSTRAINT `fk_environmental_room000` FOREIGN KEY (`dwelling_room_oid`) REFERENCES `dwelling_room` (`oid`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB DEFAULT CHARSET=utf8;


-- Dumping structure for table encompass_model.indoor_conditions_humidity
CREATE TABLE IF NOT EXISTS `indoor_conditions_humidity` (
  `oid` int(11) NOT NULL AUTO_INCREMENT,
  `dwelling_room_oid` int(11) DEFAULT NULL,
  `value` double DEFAULT NULL,
  `datetime` datetime DEFAULT NULL,
  `datetime_received` datetime DEFAULT NULL,
  PRIMARY KEY (`oid`),
  KEY `fk_environmental_room_idx` (`dwelling_room_oid`),
  CONSTRAINT `fk_environmental_room00` FOREIGN KEY (`dwelling_room_oid`) REFERENCES `dwelling_room` (`oid`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB AUTO_INCREMENT=143440 DEFAULT CHARSET=utf8;


-- Dumping structure for table encompass_model.indoor_conditions_luminance
CREATE TABLE IF NOT EXISTS `indoor_conditions_luminance` (
  `oid` int(11) NOT NULL AUTO_INCREMENT,
  `dwelling_room_oid` int(11) DEFAULT NULL,
  `value` double DEFAULT NULL,
  `datetime` datetime DEFAULT NULL,
  `datetime_received` datetime DEFAULT NULL,
  PRIMARY KEY (`oid`),
  KEY `fk_environmental_room_idx` (`dwelling_room_oid`),
  CONSTRAINT `fk_environmental_room0` FOREIGN KEY (`dwelling_room_oid`) REFERENCES `dwelling_room` (`oid`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB AUTO_INCREMENT=138759 DEFAULT CHARSET=utf8;


-- Dumping structure for table encompass_model.indoor_conditions_temperature
CREATE TABLE IF NOT EXISTS `indoor_conditions_temperature` (
  `oid` int(11) NOT NULL AUTO_INCREMENT,
  `dwelling_room_oid` int(11) DEFAULT NULL,
  `value` double DEFAULT NULL,
  `datetime` datetime DEFAULT NULL,
  `datetime_received` datetime DEFAULT NULL,
  PRIMARY KEY (`oid`),
  KEY `fk_environmental_room_idx` (`dwelling_room_oid`),
  CONSTRAINT `fk_environmental_room` FOREIGN KEY (`dwelling_room_oid`) REFERENCES `dwelling_room` (`oid`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB AUTO_INCREMENT=141761 DEFAULT CHARSET=utf8;


-- Dumping structure for table encompass_model.indoor_sensing_occupancy
CREATE TABLE IF NOT EXISTS `indoor_sensing_occupancy` (
  `oid` int(11) NOT NULL AUTO_INCREMENT,
  `dwelling_room_oid` int(11) DEFAULT NULL,
  `value` double DEFAULT NULL,
  `datetime` datetime DEFAULT NULL,
  `datetime_received` datetime DEFAULT NULL,
  PRIMARY KEY (`oid`),
  KEY `fk_environmental_room_idx` (`dwelling_room_oid`),
  CONSTRAINT `fk_environmental_room1` FOREIGN KEY (`dwelling_room_oid`) REFERENCES `dwelling_room` (`oid`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB AUTO_INCREMENT=89215 DEFAULT CHARSET=utf8;


-- Dumping structure for table encompass_model.main_lighting_type
CREATE TABLE IF NOT EXISTS `main_lighting_type` (
  `id` int(11) NOT NULL,
  `language` enum('en','it','de','gr') NOT NULL,
  `text` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`id`,`language`)
) ENGINE=InnoDB DEFAULT CHARSET=utf16;


-- Dumping structure for table encompass_model.message_feedback
CREATE TABLE IF NOT EXISTS `message_feedback` (
  `oid` int(11) NOT NULL AUTO_INCREMENT,
  `message_instance_oid` int(11) NOT NULL,
  `timestamp` datetime NOT NULL,
  `feedback` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`message_instance_oid`,`timestamp`),
  KEY `oid` (`oid`),
  CONSTRAINT `fk_message_feedback_message_instance` FOREIGN KEY (`message_instance_oid`) REFERENCES
`message_instance` (`oid`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=136 DEFAULT CHARSET=utf8;


-- Dumping structure for table encompass_model.message_instance
CREATE TABLE IF NOT EXISTS `message_instance` (
  `oid` int(11) NOT NULL AUTO_INCREMENT,
  `generic_message_oid` int(11) DEFAULT NULL,
  `user_oid` int(11) DEFAULT NULL,
  `timestamp_creation` datetime DEFAULT NULL,
  `timestamp_read` datetime DEFAULT NULL,
  `hidden` tinyint(1) DEFAULT NULL,
  `is_static` tinyint(1) DEFAULT NULL,
  `is_processed` tinyint(1) DEFAULT '0',
  PRIMARY KEY (`oid`),
  KEY `fk_tips_users_idx` (`user_oid`),
  KEY `fk_tip_instance_idx` (`generic_message_oid`),
  CONSTRAINT `fk_message_instance_generic_message` FOREIGN KEY (`generic_message_oid`) REFERENCES
`generic_message` (`oid`) ON DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT `fk_message_instance_user` FOREIGN KEY (`user_oid`) REFERENCES `user` (`oid`) ON DELETE CASCADE ON
UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=1646 DEFAULT CHARSET=utf8;


-- Dumping structure for table encompass_model.message_localization
CREATE TABLE IF NOT EXISTS `message_localization` (
  `oid` int(11) NOT NULL AUTO_INCREMENT,
  `generic_message_oid` int(11) NOT NULL,
  `language` varchar(255) NOT NULL,
  `title` varchar(45) DEFAULT NULL,
  `description` varchar(255) DEFAULT NULL,
  `image` varchar(255) DEFAULT NULL,
  `video` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`generic_message_oid`,`language`),
  KEY `oid` (`oid`),
  CONSTRAINT `FK_message_localization_generic_message` FOREIGN KEY (`generic_message_oid`) REFERENCES
`generic_message` (`oid`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=185 DEFAULT CHARSET=utf8;


-- Dumping structure for table encompass_model.meter_consumption
CREATE TABLE IF NOT EXISTS `meter_consumption` (
  `oid` int(11) NOT NULL AUTO_INCREMENT,
  `smart_meter_oid` int(11) DEFAULT NULL,
  `datetime` datetime DEFAULT NULL,
  `datetime_received` datetime DEFAULT NULL,
  `consumption` decimal(19,3) DEFAULT NULL,
  PRIMARY KEY (`oid`),
  KEY `fk_meter_consum_meter_idx` (`smart_meter_oid`),
  CONSTRAINT `fk_meter_consum_meter` FOREIGN KEY (`smart_meter_oid`) REFERENCES `smart_meter` (`oid`) ON DELETE
NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB AUTO_INCREMENT=288169 DEFAULT CHARSET=utf8;


-- Dumping structure for view encompass_model.meter_consumption_daily_avg
-- Creating temporary table to overcome VIEW dependency errors
CREATE TABLE `meter_consumption_daily_avg` (
	`user_oid` INT(11) NOT NULL,
	`daily_consumption` DECIMAL(41,3) NULL,
	`day` INT(2) NULL,
	`month` INT(2) NULL,
	`year` INT(4) NULL
) ENGINE=MyISAM;


-- Dumping structure for table encompass_model.meter_consumption_error
CREATE TABLE IF NOT EXISTS `meter_consumption_error` (
  `oid` int(11) NOT NULL AUTO_INCREMENT,
  `smart_meter_oid` int(11) DEFAULT NULL,
  `datetime` datetime DEFAULT NULL,
  `datetime_received` datetime DEFAULT NULL,
  `consumption` decimal(19,3) DEFAULT NULL,
  PRIMARY KEY (`oid`),
  KEY `fk_meter_consum_error_idx` (`smart_meter_oid`),
  CONSTRAINT `fk_meter_consum_error_smart_meter` FOREIGN KEY (`smart_meter_oid`) REFERENCES `smart_meter` (`oid`)
ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB DEFAULT CHARSET=utf8;


-- Dumping structure for view encompass_model.meter_consumption_from_to
-- Creating temporary table to overcome VIEW dependency errors
CREATE TABLE `meter_consumption_from_to` (
	`user_oid` INT(11) NOT NULL,
	`from_` DATETIME NULL,
	`to_` DATETIME NULL
) ENGINE=MyISAM;


-- Dumping structure for view encompass_model.meter_consumption_monthly_avg
-- Creating temporary table to overcome VIEW dependency errors
CREATE TABLE `meter_consumption_monthly_avg` (
	`oid` VARCHAR(36) NOT NULL COLLATE 'utf8_general_ci',
	`user_oid` INT(11) NOT NULL,
	`monthly_consumption` DECIMAL(40,2) NULL,
	`month` INT(2) NULL,
	`year` INT(4) NULL
) ENGINE=MyISAM;


-- Dumping structure for table encompass_model.meter_consumption_normalized
CREATE TABLE IF NOT EXISTS `meter_consumption_normalized` (
  `oid` int(11) NOT NULL AUTO_INCREMENT,
  `meter_consumption_oid` int(11) NOT NULL,
  `datetime_reading` datetime DEFAULT NULL,
  `total_consumption` decimal(19,3) DEFAULT NULL,
  `total_consumption_adjusted` decimal(19,3) DEFAULT NULL,
  PRIMARY KEY (`oid`),
  UNIQUE KEY `unique_reading` (`datetime_reading`),
  KEY `fk_meter_consumption_normalized_meter_consumption1_idx` (`meter_consumption_oid`),
  CONSTRAINT `fk_meter_consumption_normalized_meter_consumption1` FOREIGN KEY (`meter_consumption_oid`)
REFERENCES `meter_consumption` (`oid`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB DEFAULT CHARSET=utf16;


-- Dumping structure for view encompass_model.meter_consumption_this_month_total
-- Creating temporary table to overcome VIEW dependency errors
CREATE TABLE `meter_consumption_this_month_total` (
	`user_oid` INT(11) NOT NULL,
	`this_month_total` DECIMAL(41,3) NULL,
	`month` INT(2) NULL,
	`year` INT(4) NULL
) ENGINE=MyISAM;


-- Dumping structure for view encompass_model.meter_consumption_this_week_total
-- Creating temporary table to overcome VIEW dependency errors
CREATE TABLE `meter_consumption_this_week_total` (
	`user_oid` INT(11) NOT NULL,
	`this_week_total` DECIMAL(41,3) NULL,
	`week` INT(2) NULL,
	`year` INT(4) NULL
) ENGINE=MyISAM;


-- Dumping structure for view encompass_model.meter_consumption_today_total
-- Creating temporary table to overcome VIEW dependency errors
CREATE TABLE `meter_consumption_today_total` (
	`user_oid` INT(11) NOT NULL,
	`today_total` DECIMAL(41,3) NULL,
	`day` INT(2) NULL,
	`month` INT(2) NULL,
	`year` INT(4) NULL
) ENGINE=MyISAM;


-- Dumping structure for view encompass_model.meter_consumption_weekly_avg
-- Creating temporary table to overcome VIEW dependency errors
CREATE TABLE `meter_consumption_weekly_avg` (
	`user_oid` INT(11) NOT NULL,
	`weekly_consumption` DECIMAL(41,3) NULL,
	`week` INT(2) NULL,
	`year` INT(4) NULL
) ENGINE=MyISAM;

-- Dumping structure for table encompass_model.meter_reading
CREATE TABLE IF NOT EXISTS `meter_reading` (
  `oid` int(11) NOT NULL AUTO_INCREMENT,
  `smart_meter_oid` int(11) DEFAULT NULL,
  `datetime` datetime DEFAULT NULL,
  `datetime_received` datetime DEFAULT NULL,
  `reading` decimal(19,3) DEFAULT NULL,
  PRIMARY KEY (`oid`),
  UNIQUE KEY `unique_reading` (`datetime`,`smart_meter_oid`),
  KEY `fk_meter_reading_smart_meter` (`smart_meter_oid`),
  CONSTRAINT `fk_meter_reading_smart_meter` FOREIGN KEY (`smart_meter_oid`) REFERENCES `smart_meter` (`oid`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- Dumping structure for table encompass_model.meter_reading_error
CREATE TABLE IF NOT EXISTS `meter_reading_error` (
  `oid` int(11) NOT NULL AUTO_INCREMENT,
  `smart_meter_oid` int(11) DEFAULT NULL,
  `datetime` datetime DEFAULT NULL,
  `datetime_received` datetime DEFAULT NULL,
  `reading` decimal(19,3) DEFAULT NULL,
  PRIMARY KEY (`oid`),
  KEY `fk_meter_reading_error_idx` (`smart_meter_oid`),
  CONSTRAINT `fk_meter_reading_error_smart_meter` FOREIGN KEY (`smart_meter_oid`) REFERENCES `smart_meter` (`oid`)
ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- Dumping structure for table encompass_model.meter_reading_normalized
CREATE TABLE IF NOT EXISTS `meter_reading_normalized` (
  `oid` int(11) NOT NULL AUTO_INCREMENT,
  `meter_reading_oid` int(11) NOT NULL,
  `datetime_reading` datetime DEFAULT NULL,
  `total_consumption` decimal(19,3) DEFAULT NULL,
  `total_consumption_adjusted` decimal(19,3) DEFAULT NULL,
  PRIMARY KEY (`oid`),
  UNIQUE KEY `unique_reading` (`datetime_reading`),
  KEY `fk_meter_reading_normalized_meter_reading1_idx` (`meter_reading_oid`),
  CONSTRAINT `fk_meter_reading_normalized_meter_reading1` FOREIGN KEY (`meter_reading_oid`) REFERENCES
`meter_reading` (`oid`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB DEFAULT CHARSET=utf8;


-- Dumping structure for table encompass_model.notification_delivery
CREATE TABLE IF NOT EXISTS `notification_delivery` (
  `oid` int(11) NOT NULL AUTO_INCREMENT,
  `notification_instance_oid` int(11) NOT NULL,
  `current_priority` varchar(50) DEFAULT NULL,
  `delivery_timestamp` datetime DEFAULT NULL,
  `delivery_status` varchar(50) DEFAULT NULL,
  `active` tinyint(1) DEFAULT NULL,
  PRIMARY KEY (`oid`),
  KEY `fk_user_recommendations_user1_idx` (`notification_instance_oid`),
  CONSTRAINT `fk_notification_delivery_notification_instance` FOREIGN KEY (`notification_instance_oid`) REFERENCES
`notification_instance` (`oid`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=972 DEFAULT CHARSET=utf8;


-- Dumping structure for table encompass_model.notification_feedback
CREATE TABLE IF NOT EXISTS `notification_feedback` (
  `oid` int(11) NOT NULL AUTO_INCREMENT,
  `notification_delivery_oid` int(11) NOT NULL,
  `feedback_timestamp` datetime DEFAULT NULL,
  `feedback_status` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`oid`),
  KEY `fk_user_recommendations_user10_idx` (`notification_delivery_oid`),
  CONSTRAINT `fk_user_recommendations_user10` FOREIGN KEY (`notification_delivery_oid`) REFERENCES
`notification_delivery` (`oid`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
-- Dumping structure for table encompass_model.notification_instance
CREATE TABLE IF NOT EXISTS `notification_instance` (
  `oid` int(11) NOT NULL AUTO_INCREMENT,
  `user_oid` int(11) DEFAULT NULL,
  `notification_type_oid` int(11) DEFAULT NULL,
  `is_static` tinyint(1) DEFAULT NULL,
  `title` varchar(45) DEFAULT NULL,
  `description` varchar(255) DEFAULT NULL,
  `scheduled_priority` varchar(50) DEFAULT NULL,
  `scheduled_timestamp` datetime DEFAULT NULL,
  `scheduled_status` varchar(50) DEFAULT NULL,
  `object_id` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`oid`),
  KEY `act_user_idx` (`user_oid`),
  KEY `fk_notification_type_idx` (`notification_type_oid`),
  CONSTRAINT `act_user0` FOREIGN KEY (`user_oid`) REFERENCES `user` (`oid`) ON DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT `fk_notification_type` FOREIGN KEY (`notification_type_oid`) REFERENCES `notification_type` (`oid`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB AUTO_INCREMENT=1325 DEFAULT CHARSET=utf8;


-- Dumping structure for table encompass_model.notification_type
CREATE TABLE IF NOT EXISTS `notification_type` (
  `oid` int(11) NOT NULL AUTO_INCREMENT,
  `type` varchar(255) DEFAULT NULL,
  `default_priority` varchar(50) DEFAULT NULL,
  `duplicate_control` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`oid`),
  KEY `type` (`type`)
) ENGINE=InnoDB AUTO_INCREMENT=13 DEFAULT CHARSET=utf8;


-- Dumping structure for table encompass_model.occupancy_inference
CREATE TABLE IF NOT EXISTS `occupancy_inference` (
  `oid` int(11) NOT NULL AUTO_INCREMENT,
  `dwelling_oid` int(11) NOT NULL,
  `occupancy` int(11) DEFAULT NULL,
  `timestamp` datetime DEFAULT NULL,
  `timestamp_received` datetime DEFAULT NULL,
  PRIMARY KEY (`oid`),
  KEY `fk_Occupancy_dwelling1_idx` (`dwelling_oid`),
  CONSTRAINT `fk_Occupancy_dwelling1` FOREIGN KEY (`dwelling_oid`) REFERENCES `dwelling` (`oid`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB DEFAULT CHARSET=utf8;


-- Dumping structure for table encompass_model.on_the_fly_message_localization
CREATE TABLE IF NOT EXISTS `on_the_fly_message_localization` (
  `oid` int(11) NOT NULL AUTO_INCREMENT,
  `message_instance_oid` int(11) NOT NULL,
  `language` varchar(255) NOT NULL,
  `title` varchar(40) DEFAULT NULL,
  `description` varchar(255) DEFAULT NULL,
  `message_type` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`message_instance_oid`,`language`),
  KEY `oid` (`oid`),
  CONSTRAINT `fk_message_localization` FOREIGN KEY (`message_instance_oid`) REFERENCES `message_instance` (`oid`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=utf8;


-- Dumping structure for table encompass_model.room_comfort_inference
CREATE TABLE IF NOT EXISTS `room_comfort_inference` (
  `oid` int(11) NOT NULL AUTO_INCREMENT,
  `dwelling_room_oid` int(11) NOT NULL,
  `thermal_comfort` double DEFAULT NULL,
  `visual_comfort` double DEFAULT NULL,
  `timestamp` datetime DEFAULT NULL,
  `timestamp_received` datetime DEFAULT NULL,
  PRIMARY KEY (`oid`),
  KEY `fk_user_comfort_user10_idx` (`dwelling_room_oid`),
  CONSTRAINT `fk_user_comfort_user10` FOREIGN KEY (`dwelling_room_oid`) REFERENCES `dwelling_room` (`oid`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB AUTO_INCREMENT=21 DEFAULT CHARSET=utf8;


-- Dumping structure for table encompass_model.route_config
CREATE TABLE IF NOT EXISTS `route_config` (
  `oid` int(11) NOT NULL AUTO_INCREMENT,
  `route` varchar(255) NOT NULL DEFAULT '0',
  `source_path` varchar(255) NOT NULL DEFAULT '0',
  `error_path` varchar(255) NOT NULL DEFAULT '0',
  `md5_path` varchar(255) NOT NULL DEFAULT '0',
  `md5_check_timeout` int(11) NOT NULL DEFAULT '25' COMMENT 'time in seconds',
  `outbox_path` varchar(255) NOT NULL DEFAULT '0',
  PRIMARY KEY (`oid`)
) ENGINE=InnoDB AUTO_INCREMENT=7 DEFAULT CHARSET=utf8;


-- Dumping structure for table encompass_model.schedule_dwelling
CREATE TABLE IF NOT EXISTS `schedule_dwelling` (
  `oid` int(11) NOT NULL AUTO_INCREMENT,
  `dwelling_oid` int(11) NOT NULL,
  `time_open` time DEFAULT NULL,
  `time_close` time DEFAULT NULL,
  `time_open_weekend` time DEFAULT NULL,
  `time_close_weekend` time DEFAULT NULL,
  PRIMARY KEY (`oid`),
  KEY `fk_schedule_dwelling_dwelling1_idx` (`dwelling_oid`),
  CONSTRAINT `fk_schedule_dwelling_dwelling1` FOREIGN KEY (`dwelling_oid`) REFERENCES `dwelling` (`oid`) ON DELETE
NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB DEFAULT CHARSET=utf8;


-- Dumping structure for table encompass_model.semaphore_log
CREATE TABLE IF NOT EXISTS `semaphore_log` (
  `oid` int(11) NOT NULL AUTO_INCREMENT,
  `timestamp` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `component` varchar(50) NOT NULL,
  `process` varchar(50) NOT NULL,
  `status_code` int(11) NOT NULL,
  PRIMARY KEY (`oid`)
) ENGINE=InnoDB AUTO_INCREMENT=12 DEFAULT CHARSET=utf16;


-- Dumping structure for table encompass_model.simple_device_instance
CREATE TABLE IF NOT EXISTS `simple_device_instance` (
  `oid` int(11) NOT NULL AUTO_INCREMENT,
  `device_type_oid` int(11) DEFAULT NULL,
  `dwelling_room_oid` int(11) DEFAULT NULL,
  `number` int(11) DEFAULT NULL,
  `nominal_power` decimal(19,3) DEFAULT NULL,
  PRIMARY KEY (`oid`),
  KEY `fk_simple_device_instance_devi` (`device_type_oid`),
  KEY `fk_simple_device_instance_hous_idx` (`dwelling_room_oid`),
  CONSTRAINT `fk_simple_device_instance_devi` FOREIGN KEY (`device_type_oid`) REFERENCES `device_type` (`oid`),
  CONSTRAINT `fk_simple_device_instance_hous` FOREIGN KEY (`dwelling_room_oid`) REFERENCES `dwelling_room` (`oid`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

-- Dumping structure for table encompass_model.smart_meter
CREATE TABLE IF NOT EXISTS `smart_meter` (
  `oid` int(11) NOT NULL AUTO_INCREMENT,
  `dwelling_oid` int(11) DEFAULT NULL,
  `smart_meter_id` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`oid`),
  KEY `fk_smart_meter_building_idx` (`dwelling_oid`),
  CONSTRAINT `fk_smart_meter_building` FOREIGN KEY (`dwelling_oid`) REFERENCES `dwelling` (`oid`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=175 DEFAULT CHARSET=utf8;

-- Dumping structure for table encompass_model.user
CREATE TABLE IF NOT EXISTS `user` (
  `oid` int(11) NOT NULL AUTO_INCREMENT,
  `dwelling_oid` int(11) NOT NULL,
  `username` varchar(255) DEFAULT NULL,
  `password` varchar(255) DEFAULT NULL,
  `email` varchar(255) DEFAULT NULL,
  `birth_date` datetime DEFAULT NULL,
  `internal` bit(1) DEFAULT NULL,
  `type` varchar(255) DEFAULT NULL,
  `language` varchar(255) DEFAULT NULL,
  `title` varchar(45) DEFAULT NULL,
  `saving_goal_money` int(11) DEFAULT NULL,
  `saving_goal_fun` int(11) DEFAULT NULL,

  `saving_goal_environmental` int(11) DEFAULT NULL,

  `firebasegroup_id` varchar(255) DEFAULT NULL,

  `firebasegroup_name` varchar(255) DEFAULT NULL,

  `motivation` int(11) DEFAULT NULL,

  PRIMARY KEY (`oid`),

  KEY `fk_user_dwelling1_idx` (`dwelling_oid`),

  CONSTRAINT `fk_user_dwelling1` FOREIGN KEY (`dwelling_oid`) REFERENCES `dwelling` (`oid`) ON DELETE CASCADE ON UPDATE CASCADE

) ENGINE=InnoDB AUTO_INCREMENT=176 DEFAULT CHARSET=utf8;


-- Dumping structure for table encompass_model.user_comfort_feedback

CREATE TABLE IF NOT EXISTS `user_comfort_feedback` (

  `oid` int(11) NOT NULL AUTO_INCREMENT,

  `user_oid` int(11) NOT NULL,

  `thermal_comfort` double DEFAULT NULL,

  `visual_comfort` double DEFAULT NULL,

  `timestamp` datetime DEFAULT NULL,

  `timestamp_received` datetime DEFAULT NULL,

  `feedback_type` varchar(255) DEFAULT NULL,

  PRIMARY KEY (`oid`),

  KEY `fk_user_comfort_user1_idx` (`user_oid`),

  CONSTRAINT `fk_user_comfort_user1` FOREIGN KEY (`user_oid`) REFERENCES `user` (`oid`) ON DELETE NO ACTION ON UPDATE NO ACTION

) ENGINE=InnoDB AUTO_INCREMENT=70 DEFAULT CHARSET=utf8;


-- Dumping structure for table encompass_model.user_group

CREATE TABLE IF NOT EXISTS `user_group` (

  `group_oid` int(11) NOT NULL,

  `user_oid` int(11) NOT NULL,

  KEY `fk_user_group_group` (`group_oid`),

  KEY `fk_user_group_user1_idx` (`user_oid`),

  CONSTRAINT `fk_user_group_group` FOREIGN KEY (`group_oid`) REFERENCES `group` (`oid`) ON DELETE CASCADE ON UPDATE CASCADE,

  CONSTRAINT `fk_user_group_user1` FOREIGN KEY (`user_oid`) REFERENCES `user` (`oid`) ON DELETE CASCADE ON UPDATE CASCADE

) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
-- Dumping structure for table encompass_model.user_preference

CREATE TABLE IF NOT EXISTS `user_preference` (

  `oid` int(11) NOT NULL AUTO_INCREMENT,

  `user_oid` int(11) DEFAULT NULL,

  `language` varchar(255) DEFAULT NULL,

  `timezone` int(11) DEFAULT NULL,

  `receive_notification` tinyint(1) DEFAULT NULL,

  `notification_sent_at_once` int(11) DEFAULT NULL,

  PRIMARY KEY (`oid`),

  UNIQUE KEY `oid_UNIQUE` (`oid`),

  KEY `fk_pref_users_idx` (`user_oid`),

  CONSTRAINT `fk_pref_users` FOREIGN KEY (`user_oid`) REFERENCES `user` (`oid`) ON DELETE CASCADE ON UPDATE
CASCADE

) ENGINE=InnoDB AUTO_INCREMENT=175 DEFAULT CHARSET=utf8;


-- Dumping structure for table encompass_model.user_preference_daytime

CREATE TABLE IF NOT EXISTS `user_preference_daytime` (

  `oid` int(11) NOT NULL AUTO_INCREMENT,

  `user_preference_oid` int(11) DEFAULT NULL,

  `preferred_day` varchar(50) DEFAULT NULL,

  `preferred_time_start` varchar(50) DEFAULT NULL,

  `preferred_time_end` varchar(50) DEFAULT NULL,

  PRIMARY KEY (`oid`),

  UNIQUE KEY `oid_UNIQUE` (`oid`),

  KEY `fk_pref_daytime_idx` (`user_preference_oid`),

  CONSTRAINT `fk_pref_daytime` FOREIGN KEY (`user_preference_oid`) REFERENCES `user_preference` (`oid`) ON DELETE
CASCADE ON UPDATE CASCADE

) ENGINE=InnoDB AUTO_INCREMENT=181 DEFAULT CHARSET=utf8;


-- Dumping structure for table encompass_model.user_profile

CREATE TABLE IF NOT EXISTS `user_profile` (

  `oid` int(11) NOT NULL AUTO_INCREMENT,

  `user_oid` int(11) DEFAULT NULL,

  `no_of_adults_older_than_16` int(11) DEFAULT NULL,

  `no_of_kids_younger_than_16` int(11) DEFAULT NULL,
  `no_of_pets` int(11) DEFAULT NULL,
  `living_since_may_2017` tinyint(1) DEFAULT NULL,
  `dwelling_type_oid` int(11) DEFAULT NULL,
  `number_of_rooms` int(11) DEFAULT NULL,
  `heating_type_oid` int(11) DEFAULT NULL,
  `heating_source_type_oid` int(11) DEFAULT NULL,
  `heat_pump` tinyint(1) DEFAULT NULL,
  `water_boiler` tinyint(1) DEFAULT NULL,
  `electric_car` tinyint(1) DEFAULT NULL,
  `main_lighting_type_oid` int(11) DEFAULT NULL,
  `AC` tinyint(1) DEFAULT NULL,
  `electric_oven` tinyint(1) DEFAULT NULL,
  `microwave` tinyint(1) DEFAULT NULL,
  `electric_hot_plates` tinyint(1) DEFAULT NULL,
  `electric_kettle` tinyint(1) DEFAULT NULL,
  `coffee_machine` tinyint(1) DEFAULT NULL,
  `vacuum_cleaner` tinyint(1) DEFAULT NULL,
  `dishwasher` tinyint(1) DEFAULT NULL,
  `washing_machine_existence` tinyint(1) DEFAULT NULL,
  `washing_machine_energy_info` varchar(255) DEFAULT NULL,
  `washing_machine_shared` tinyint(1) DEFAULT NULL,
  `tumble_dryer_existence` tinyint(1) DEFAULT NULL,
  `tumble_dryer_energy_info` varchar(255) DEFAULT NULL,
  `tumble_dryer_shared` tinyint(1) DEFAULT NULL,
  `dehumidifier` tinyint(1) DEFAULT NULL,
  `fridge` int(11) DEFAULT NULL,
  `freezer` int(11) DEFAULT NULL,
  `tv_set` int(11) DEFAULT NULL,
  `hi_fi` int(11) DEFAULT NULL,
  `desktop_computer` int(11) DEFAULT NULL,
  `laptop_computer` int(11) DEFAULT NULL,
  `gaming_set` int(11) DEFAULT NULL,
  `energy_saving_motivation` varchar(255) DEFAULT NULL,
  `occupancy_during_day_weekdays` tinyint(1) DEFAULT NULL,
  `occupancy_during_night_weekdays` tinyint(1) DEFAULT NULL,
  `occupancy_during_day_weekend` tinyint(1) DEFAULT NULL,
  `occupancy_during_night_weekend` tinyint(1) DEFAULT NULL,
  `desired_thermal_comfort` int(11) DEFAULT NULL,
  `desired_visual_comfort` int(11) DEFAULT NULL,
  PRIMARY KEY (`oid`),
  KEY `fk_user_recom_settings_idx` (`user_oid`),
  KEY `fk_dwelling_type_idx` (`dwelling_type_oid`),
  KEY `fk_heating_type_idx` (`heating_type_oid`),
  KEY `fk_heating_source_type_idx` (`heating_source_type_oid`),
  KEY `fk_main_lighting_type_idx` (`main_lighting_type_oid`),
  CONSTRAINT `fk_dwelling_type` FOREIGN KEY (`dwelling_type_oid`) REFERENCES `dwelling_type` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION,
  CONSTRAINT `fk_heating_source` FOREIGN KEY (`heating_source_type_oid`) REFERENCES `heating_source_type` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION,
  CONSTRAINT `fk_heating_type` FOREIGN KEY (`heating_type_oid`) REFERENCES `heating_type` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION,
  CONSTRAINT `fk_main_lighting_type` FOREIGN KEY (`main_lighting_type_oid`) REFERENCES `main_lighting_type` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION,
  CONSTRAINT `fk_user_recom_settings` FOREIGN KEY (`user_oid`) REFERENCES `user` (`oid`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=174 DEFAULT CHARSET=utf8;


-- Dumping structure for table encompass_model.weather_condition
CREATE TABLE IF NOT EXISTS `weather_condition` (
  `oid` int(11) NOT NULL AUTO_INCREMENT,
  `district_oid` int(11) DEFAULT NULL,
  `date_start` date DEFAULT NULL,
  `date_end` date DEFAULT NULL,
  `rain_fall` decimal(19,2) DEFAULT NULL,
  `avg_temperature` decimal(19,2) DEFAULT NULL,
  `avg_co2` decimal(19,2) DEFAULT NULL,
  `avg_humidity` decimal(19,2) DEFAULT NULL,
  PRIMARY KEY (`oid`),
  KEY `fk_weather_condition_district` (`district_oid`),
  CONSTRAINT `fk_weather_condition_district` FOREIGN KEY (`district_oid`) REFERENCES `district` (`oid`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;



-- Dumping structure for view encompass_model.get_tips
-- Removing temporary table and create final VIEW structure

DROP TABLE IF EXISTS `get_tips`;

CREATE ALGORITHM=UNDEFINED DEFINER=`admin`@`%` SQL SECURITY DEFINER VIEW `get_tips` AS select `mi`.`user_oid`
AS `user_oid`,`gm`.`oid` AS `id`,`ml`.`title` AS `name`,'' AS `header`,`ml`.`description` AS `body`,`ml`.`image` AS
`image`,`ml`.`video` AS `video`,`mi`.`oid` AS `instance`,`mf2`.`feedback` AS `feedback`,`mi`.`is_static` AS
`static`,`mi`.`timestamp_creation` AS `timestamp_creation` from ((((`generic_message` `gm` join `message_instance` `mi`
on((`mi`.`generic_message_oid` = `gm`.`oid`))) join `message_localization` `ml` on(((`ml`.`generic_message_oid` = `gm`.`oid`)
and (`mi`.`is_static` = 1)))) join `user_preference` `up` on(((`up`.`user_oid` = `mi`.`user_oid`) and (`up`.`language` =
`ml`.`language`)))) left join `get_tips_with_feedback` `mf2` on((`mf2`.`message_instance_oid` = `mi`.`oid`))) where
(`mi`.`hidden` = 0) union select `mi`.`user_oid` AS `user_oid`,`gm`.`oid` AS `id`,`ofml`.`title` AS `name`,'' AS
`header`,`ofml`.`description` AS `body`,NULL AS `image`,NULL AS `video`,`mi`.`oid` AS `instance`,`mf2`.`feedback` AS
`feedback`,`mi`.`is_static` AS `static`,`mi`.`timestamp_creation` AS `timestamp_creation` from ((((`generic_message` `gm` join
`message_instance` `mi` on((`mi`.`generic_message_oid` = `gm`.`oid`))) join `on_the_fly_message_localization` `ofml`
on(((`ofml`.`message_instance_oid` = `gm`.`oid`) and (`mi`.`is_static` = 0)))) join `user_preference` `up` on(((`up`.`user_oid` =
`mi`.`user_oid`) and (`up`.`language` = `ofml`.`language`)))) left join `get_tips_with_feedback` `mf2`
on((`mf2`.`message_instance_oid` = `mi`.`oid`))) where (`mi`.`hidden` = 0);

-- Dumping structure for view encompass_model.get_tips_with_feedback

-- Removing temporary table and create final VIEW structure

DROP TABLE IF EXISTS `get_tips_with_feedback`;

CREATE ALGORITHM=UNDEFINED DEFINER=`admin`@`%` SQL SECURITY DEFINER VIEW `get_tips_with_feedback` AS select
`mf`.`oid` AS `oid`,`mf`.`feedback` AS `feedback`,`mf`.`message_instance_oid` AS `message_instance_oid` from
(`message_feedback` `mf` join `message_instance` `mi` on(((`mf`.`message_instance_oid` = `mi`.`oid`) and
(`mf`.`message_instance_oid` = `mi`.`generic_message_oid`)))) where `mf`.`oid` in (select `mf`.`oid` from `message_feedback`
`mf` where `mf`.`timestamp` in (select max(`mf`.`timestamp`) from `message_feedback` `mf` group by
`mf`.`message_instance_oid`)) order by `mf`.`timestamp` desc;

-- Dumping structure for view encompass_model.meter_consumption_daily_avg

-- Removing temporary table and create final VIEW structure

DROP TABLE IF EXISTS `meter_consumption_daily_avg`;

CREATE ALGORITHM=UNDEFINED DEFINER=`admin`@`*` SQL SECURITY DEFINER VIEW `meter_consumption_daily_avg` AS
select `u`.`oid` AS `user_oid`,sum(`mc`.`consumption`) AS `daily_consumption`,dayofmonth(`mc`.`datetime`) AS
`day`,month(`mc`.`datetime`) AS `month`,year(`mc`.`datetime`) AS `year` from (((`meter_consumption` `mc` join `smart_meter`
`sm` on((`mc`.`smart_meter_oid` = `sm`.`oid`))) join `dwelling` `d` on((`sm`.`dwelling_oid` = `d`.`oid`))) join `user` `u`
on((`d`.`oid` = `u`.`dwelling_oid`))) where (cast(`mc`.`datetime` as date) <= cast((now() + interval -(1) day) as date)) group by
`u`.`oid`,dayofmonth(`mc`.`datetime`),month(`mc`.`datetime`),year(`mc`.`datetime`) order by
`u`.`oid`,year(`mc`.`datetime`),month(`mc`.`datetime`),dayofmonth(`mc`.`datetime`);


-- Dumping structure for view encompass_model.meter_consumption_from_to

-- Removing temporary table and create final VIEW structure

DROP TABLE IF EXISTS `meter_consumption_from_to`;

CREATE ALGORITHM=UNDEFINED DEFINER=`admin`@`%` SQL SECURITY DEFINER VIEW `meter_consumption_from_to` AS
select `u`.`oid` AS `user_oid`,min(`mc`.`datetime`) AS `from_`,max(`mc`.`datetime`) AS `to_` from (((`meter_consumption` `mc`
join `smart_meter` `sm` on((`mc`.`smart_meter_oid` = `sm`.`oid`))) join `dwelling` `d` on((`sm`.`dwelling_oid` = `d`.`oid`))) join
`user` `u` on((`d`.`oid` = `u`.`dwelling_oid`))) where (cast(`mc`.`datetime` as date) <= cast((now() + interval -(1) day) as date))
group by `u`.`oid`;

-- Dumping structure for view encompass_model.meter_consumption_monthly_avg

-- Removing temporary table and create final VIEW structure

DROP TABLE IF EXISTS `meter_consumption_monthly_avg`;
```

```
CREATE ALGORITHM=UNDEFINED DEFINER=`admin`@`%` SQL SECURITY DEFINER VIEW
`meter_consumption_monthly_avg` AS select uuid() AS `oid`,`u`.`oid` AS `user_oid`,truncate(sum(`mc`.`consumption`),2) AS
`monthly_consumption`,month(`mc`.`datetime`) AS `month`,year(`mc`.`datetime`) AS `year` from (((`meter_consumption` `mc`
join `smart_meter` `sm` on((`mc`.`smart_meter_oid` = `sm`.`oid`))) join `dwelling` `d` on((`sm`.`dwelling_oid` = `d`.`oid`))) join
`user` `u` on((`d`.`oid` = `u`.`dwelling_oid`))) where (cast(`mc`.`datetime` as date) <= cast((now() + interval -(1) day) as date))
group by `u`.`oid`,month(`mc`.`datetime`),year(`mc`.`datetime`);

-- Dumping structure for view encompass_model.meter_consumption_this_month_total

-- Removing temporary table and create final VIEW structure

DROP TABLE IF EXISTS `meter_consumption_this_month_total`;

CREATE ALGORITHM=UNDEFINED DEFINER=`admin`@`%` SQL SECURITY DEFINER VIEW
`meter_consumption_this_month_total` AS select `u`.`oid` AS `user_oid`,sum(`mc`.`consumption`) AS
`this_month_total`,month((now() - interval 1 day)) AS `month`,year(now()) AS `year` from (((`meter_consumption` `mc` join
`smart_meter` `sm` on((`mc`.`smart_meter_oid` = `sm`.`oid`))) join `dwelling` `d` on((`sm`.`dwelling_oid` = `d`.`oid`))) join
`user` `u` on((`d`.`oid` = `u`.`dwelling_oid`))) where ((month(`mc`.`datetime`) = month((now() - interval 1 day))) and
(year(`mc`.`datetime`) = year((now() - interval 1 day))) and (cast(`mc`.`datetime` as date) <= cast((now() + interval -(1) day)
as date))) group by `u`.`oid`;

-- Dumping structure for view encompass_model.meter_consumption_this_week_total

-- Removing temporary table and create final VIEW structure

DROP TABLE IF EXISTS `meter_consumption_this_week_total`;

CREATE ALGORITHM=UNDEFINED DEFINER=`admin`@`%` SQL SECURITY DEFINER VIEW
`meter_consumption_this_week_total` AS select `u`.`oid` AS `user_oid`,sum(`mc`.`consumption`) AS
`this_week_total`,week((now() - interval 1 day),0) AS `week`,year((now() - interval 1 day)) AS `year` from
(((`meter_consumption` `mc` join `smart_meter` `sm` on((`mc`.`smart_meter_oid` = `sm`.`oid`))) join `dwelling` `d`
on((`sm`.`dwelling_oid` = `d`.`oid`))) join `user` `u` on((`d`.`oid` = `u`.`dwelling_oid`))) where ((week(`mc`.`datetime`,7) =
week(now(),7)) and (year(`mc`.`datetime`) = year(now())) and (cast(`mc`.`datetime` as date) <= cast((now() + interval -(1)
day) as date))) group by `u`.`oid`;

-- Dumping structure for view encompass_model.meter_consumption_today_total

-- Removing temporary table and create final VIEW structure

DROP TABLE IF EXISTS `meter_consumption_today_total`;

CREATE ALGORITHM=UNDEFINED DEFINER=`admin`@`%` SQL SECURITY DEFINER VIEW `meter_consumption_today_total`
AS select `u`.`oid` AS `user_oid`,sum(`mc`.`consumption`) AS `today_total`,dayofmonth((now() - interval 1 day)) AS
`day`,month((now() - interval 1 day)) AS `month`,year((now() - interval 1 day)) AS `year` from (((`meter_consumption` `mc`
join `smart_meter` `sm` on((`mc`.`smart_meter_oid` = `sm`.`oid`))) join `dwelling` `d` on((`sm`.`dwelling_oid` = `d`.`oid`))) join
`user` `u` on((`d`.`oid` = `u`.`dwelling_oid`))) where (cast(`mc`.`datetime` as date) = cast((now() + interval -(1) day) as date))
group by `u`.`oid`;

-- Dumping structure for view encompass_model.meter_consumption_weekly_avg

-- Removing temporary table and create final VIEW structure

DROP TABLE IF EXISTS `meter_consumption_weekly_avg`;

CREATE ALGORITHM=UNDEFINED DEFINER=`admin`@`%` SQL SECURITY DEFINER VIEW `meter_consumption_weekly_avg`
AS select `u`.`oid` AS `user_oid`,sum(`mc`.`consumption`) AS `weekly_consumption`,week(`mc`.`datetime`,0) AS
`week`,year(`mc`.`datetime`) AS `year` from (((`meter_consumption` `mc` join `smart_meter` `sm` on((`mc`.`smart_meter_oid`
= `sm`.`oid`))) join `dwelling` `d` on((`sm`.`dwelling_oid` = `d`.`oid`))) join `user` `u` on((`d`.`oid` = `u`.`dwelling_oid`))) where
(cast(`mc`.`datetime` as date) <= cast((now() + interval -(1) day) as date)) group by
`u`.`oid`,week(`mc`.`datetime`,0),year(`mc`.`datetime`);

/*!40101 SET SQL_MODE=IFNULL(@OLD_SQL_MODE, '') */;

/*!40014 SET FOREIGN_KEY_CHECKS=IF(@OLD_FOREIGN_KEY_CHECKS IS NULL, 1, @OLD_FOREIGN_KEY_CHECKS) */;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
```

REFERENCES

Git for Windows, Retrieved from https://gitforwindows.org/

Apache, Maven. (n.d.). Retrieved from https://maven.apache.org/

IBM, Service Oriented Analysis and Design. (n.d.). Retrieved from https://www.ibm.com/developerworks/library/ws-soad1/index.html

Pivotal Software, SpringBoot. (n.d.). Retrieved from Spring: https://projects.spring.io/spring-boot/

RedHat, Hibernate ORM. (n.d.). Retrieved from http://hibernate.org/orm/

RM-ODP. (n.d.). Retrieved from http://www.rm-odp.net/

SmartBear Software, Swagger. (n.d.). Retrieved from https://swagger.io/

Zachman, J. A. (n.d.). Retrieved from https://www.zachman.com/about-the-zachman-framework