NYU Tandon School of Engineering
Computer Science and Engineering
CS6083, Fall 2023

November 15, 2023

Project #1 (due 12/1)

Note: In this first part of the project, you will focus on designing a suitable relational schema that can be used to store the data in the system. In the second part of the project, you will then build a web-accessible frontend that will allow users to use the service via their browsers. The second part of the project builds on top of this first part so you cannot skip this part of the project. This project may be done in teams of two students. You will not be allowed to change your team partner, or change team project to individual or vice versa. For this first part, you should use a standard relational DBMS, such as PostgreSQL, MySQL, SQL Server, or Oracle DB for all steps – the issue of what framework to use to make things web accessible will only come up in the second part of the project.

Project Overview:
In this project, you will design and develop a part of a Smart Home Energy Management System (SHEMS), that is, a system that helps homeowners efficiently manage their energy consumption and reduce electricity bills. A SHEMS integrates with various smart electrical devices, such as AC, washer, dryer, refrigerator, oven, and lights, to monitor and control energy usage. In this project, we focus on the part of the system that stores past energy usage, thus allowing a user to understand their past energy usage and costs, and how they relate to their various appliances and appliance settings. We will not try to model how the user can control future energy use and cost by, for example, setting schedules for their appliances, or avoiding running appliances at times when energy prices peak – these issues are beyond the scope of what you should do.

Problem Details:
We assume that the SHEMS database is deployed by an energy provider such as Con Edison, and its customers can then connect to the system. Customers need to be able to sign up for the system, by providing certain information such as their name and billing address. Each customer account could have several *service locations*, that is, properties receiving electrical service, and a customer should be able to add service locations to their account. (For example, a customer could have one service location at their primary residence, one at a vacation property, and maybe one at a rental property they own.) For each service location, we need to store the complete address, including the unit number in the building, the date when they took over this location (say, by moving there or buying it), and the square footage, number of bedrooms, and number of occupants of the unit (to allow energy usage comparisons across similar units).

For each service location, customers can then enroll devices, such as AC system, dryer, lights, or refrigerator. We assume that these are all *smart devices* that are able to send certain information, such as their current settings, or certain events, or their current energy usage, to the SHEMS database, as specified further below. When a user enrolls a new device, they have to specify the type (e.g., AC system, or light, or refrigerator), and the model number (e.g., GE Cafe 400 refrigerator, or Samsung QB800 electric dryer). The database will probably have prestored a certain number of models and their properties, and customers then select their model from a list provided by the system.

Note that a customer could have several appliances of the same type and model at the same location, such as several smart lights of the same model. Thus, you might have to assign an ID to each enrolled device to uniquely identify it.

Each device that is enrolled will then send certain data to the SHEMS database. One type of such data are events such as when a device is switched on and off, or when a setting is changed (e.g, the temperature setting for an AC, or the brightness on a dimmable smart light), or even when a refrigerator door is opened or closed. Another type would be information on energy consumption – we assume that when switched on, each device will send information about its recent energy use, in kwh (kilowatt hours), at regular intervals, say once every 5 minutes and also whenever a device is switched off. For each such data item, we need to store the device ID, a time stamp, the label of the event (e.g., “switched on”, “door opened”, “door closed”, “temp lowered”, or “energy use”), and a value (e.g, “78” if the label was “temp lowered” and the temperature was lowered to 78 degrees, or “0.23” if the label was “energy use” and the device used 0.23 kwh in the last 5 minutes. Note that the types of labels and values that are generated by a device would probably be specified as part of the information that is prestored for each model, but you do not have to model how this is prestored, or to cross reference the generated labels with those stored with the model. You also do not have to worry about how a device is initially paired with the database when the user enrolls it, or how the data items are sent by the devices to the database and then stored. We assume that all devices are smart devices, so there is no additional energy usage beyond those devices that you have to be concerned about.

Finally, energy prices per kwh can vary over time, and can also vary by location, and the system has to store this information. In particular, we assume that prices can change on an hourly basis, and that prices can depend on the zip code of the service location. This allows us to compute the energy cost of, say, washing a load of clothes at 3pm on Thursday last week, based on the energy prices in the neighborhood at that time.

In the second part of this project, users should be able to look at this data using a web browser, in order to better understand their past energy usage and costs. The details will be provided as part of the handout for Project #2. But you could imagine functionality such as graphs or tables showing a user’s energy usage and cost per week and per device, reports showing devices with much higher energy use than in previous months, data aggregations across all users showing how frequently opening a refrigerator door results in higher energy costs, or estimates of cost savings available by running the washer and dryer at non-peak hours with cheaper energy.

Project Steps and Deliverables:
In the following, we describe the suggested steps you take for this project, and the associated deliverables. You should approach this project like one of the design problems in the homeworks, except that the schema may end up being a bit more complicated. You should spend some time carefully designing sample data that allows you to test some of the functionality. The suggested steps are as follows:

(a) Design, justify, and create an appropriate relational database schema for the above scenario. Make sure your schema is space efficient, and suitably normalized. Show an ER diagram of your design, and a translation into relational format. Identify keys and foreign key constraints. Provide a short discussion of any assumptions that you made in your design, and how they impact the model.

(b) Use a relational database system to create the schema, together with key, foreign key, and other constraints.

(c) Write SQL queries for the following tasks:

1. List all enrolled devices with their total energy consumption in the last 24 hours, for a specific customer identified by customer ID.

2. Calculate the average monthly energy consumption per device type, for the month of August 2022, considering only devices that have been on (i.e., reported data) at least once during that month.

3. Identify cases where a refrigerator door was left open for more than 30 minutes. Output the date and time, the service location, the device ID, and the refrigerator model.

4. Calculate the total energy cost for each service location during August 2022, considering the hourly changing energy prices based on zip code.

5. For each service location, compute its total energy consumption during August 2022, as a percentage of the average total energy consumption during the same time of other service locations that have a similar square footage (meaning, at most 5% higher or lower square footage). Thus, you would output 150% if a service location with 1000 sqft had 50% higher energy consumption than the average of other service locations that have between 950 and 1050 sqft.

6. Identify service location(s) that had the highest percentage increase in energy consumption between August and September of 2022.

(d) Populate your database with sample data and test the queries you've written in part (c). Ensure the data is meaningful and allows for various test cases for the above queries. Make sure each queries returns at least one result. Show your test data in concise tabular form on one or two pages.

(e) Document and log your design and testing appropriately. Submit a well-documented description and justification of your entire design, including ER diagrams, tables, constraints, queries, procedures, and tests on sample data. Your documentation should be a comprehensive paper, including introduction, explanations, ER and other diagrams, and more, spanning 10-15 pages.