Monitoring and Controlling Electricity Consumption Application for Smart Buildings

Maria Irène CĂLINOIU, Simona-Vasilica OPREA
Department of Economic Informatics and Cybernetics
The Bucharest University of Economic Studies, Romania
calinoiumaria@yahoo.com, simona.oprea@csie.ase.ro

In the context of global concern about climate change and energy security, saving and optimizing electricity usage has become a priority of the European Union's energy policy. In the member states, households consume about 27% of the total electricity used [1]. *Thus, controlling electricity consumption remains an important objective for this sector as well. Developing a reliable application for electricity monitoring and optimization is a way to reduce individual consumer consumption, as well as a solution to avoid overloading the distribution network.*

Keywords: *electric energy consumption feedback, energy literacy, consumer profile, shifting optimization algorithm, mobile application design*

1 Introduction

Electric energy is a basic need, whose importance is seldom acknowledged. In the absence of an efficient, real-time monitoring system, consumers cannot perceive energy as an essential aspect, whose usage depends strictly on their habits and way of life.

Energy literacy is defined as "the understanding of energy concepts necessary to make informed decisions on energy use at both individual and social levels". [2]. This education can be formal, obtained during the schooling process, or acquired through everyday experiences.

In order to make the concept of energy more transparent, information concerning energy consumption should be made available to all consumers, with the purpose of avoiding waste and properly managing usage where the demand is bigger than the offer. Thus, consumers will be made aware of their behavior's direct impact on total consumption, usage costs, and the environment. Consequently, they will be more motivated to adapt their lifestyle in order to alter consumption habits.

Feedback, known as the process of obtaining information regarding behavior that can be used as a basis for improvement, can help increase energy literacy, therefore impacting consumer behavior [3].

Currently available technology offers the means for analyzing usage patterns. Among those devices, Smart Meters (SM), Home Energy Displays or In-Home Displays (IHD) are the most common. SM are advanced metering devices capable of recording consumption in detail and communicating it through a network to the provider in order to monitor it and simplify the billing process. Installing SM presents multiple advantages, among which: offering feedback in real time, eliminating the need to be read periodically by an employee, discouraging fraud, remotely controlling the system, and quickly identifying malfunctions [4]. IHD are modern gadgets that show more detailed information regarding consumption data in real time, by measuring data recorded by a standard meter, or by receiving data from SM.

The literature concerning feedback on energy consumption shows that, despite the initial attraction towards these devices, consumers eventually lose interest if the data provided is not considered relevant, intuitive, readable, and understandable. Moreover, the studies stress the importance of accessibility, and show that, in order to be effective, the information presented needs to be displayed somewhere the user finds convenient and within reach.

Web and mobile software products complement the physical devices designed to monitor consumption, offering flexibility and interactivity, and filtering out information deemed unimportant depending on user preferences and on relatable motivational factors.

Differential tariffs, also known as *Time-of-use tariffs*, or **ToU**, work by establishing varying costs per kWh based on the time interval during which usage takes place. They represent an alternative to fixed tariffs and could offer advantages to both consumers and energy providers.

ToUs allow optimizing consumption distribution by discouraging consumption at peak hours, mainly by raising the energy costs for certain busier time intervals. To encourage the desired behavior changes, energy companies offer low tariffs outside of peak hours, e.g. at nighttime. Implementing these tariffs is possible only through smart grids, where consumers use SM that record their usage dynamically. Through such technological advances, the former, more centralized approach is transformed into a dynamic process, adapted to client demand, known as **Demand-Side Management** (DSM) [5].

2 Objectives

The mobile application described in this article, called **Watt Manager**, is destined to put the aforementioned concepts in practice. It provides users with quick access to current and historical consumption data, allowing them to track and to optimize their usage. Consumption is monitored through goal setting, optimized by scheduling appliances in order to minimize the consumption peak, and visualized through interactive and concise diagrams and reports. Moreover, it aims to raise awareness regarding household usage habits, and to increase energy literacy.

What sets Watt Manager apart from other consumption monitoring applications is the increased focus on the consumer, and the flexibility regarding user needs. It is customized from all points of view, from units of measurement to active features, giving the user full control over the information provided. Thus, it is suitable for all household members, regardless of their age or level of involvement.

A survey completed at signup generates a consumer profile, the basis on which the app is run. This method is more effective in registering the client's preferences than a bland settings page, because it captures the user's interest, and it helps establish their motivational factors and determine their level of adaptability. Therefore, the application is customized according to the client's billing method and needs, displaying only the fitting information.

The application promotes education in the energy domain through a **Trivia** game. The game's goal is to develop the clients' knowledge and to raise awareness regarding the impact of energy consumption on the environment. Moreover, a feature that schedules appliances, based on the restrictions imposed by the client, offers better control of energy usage.

However, drastic lifestyle changes are not the app's objective. Instead, it aims to obtain gradual shifts in behavior, varying from one household to another and based on habits and consumer flexibility.

Through implementing a **shifting optimization algorithm**, the electricity consumption curve is flattened, generating a more homogenous grid load throughout the day [6]. The scheduling algorithm must take both personal and natural restrictions into account. Such restrictions could refer to certain dependencies between appliances, or to particularities of the programmable devices considered, such as whether they can be scheduled with interruptions or not.

Watt Manager eliminates the uncertainty concerning household energy consumption by displaying usage in real time. Based on the detailed information provided by the application, the clients can visualize their total usage, or grouped by category, by time interval, or by individual meter, helping them become more aware of the way their habits impact their consumption.

In addition, the app allows users to monitor consumption through goals. It also signals tariff details in the case of ToU tariffs. These elements offer the necessary information to identify the opportunities for achieving behavioral changes, while also motivating users to reduce consumption.
The app keeps a record of the household's bills, offering the possibility to accessing all the data concerning energy consumption on a single platform. The graphs and the reports generated allow a detailed analysis, depending on the type of graph, the level of aggregation, the measurement unit, and the time interval selected by the consumer.
The Trivia game offers users a chance to find out more about the energy field and, as a result, improve their consumption habits.

3 Designing the Application

Watt Manager is designed in order to facilitate the implementation of the features described above.
The advantages of a relational database, such as the well-defined relationships between its entities and the possibility to eliminate redundancy, make it the chosen format for the app's database. Its logical schema is presented below, in Figure 1.

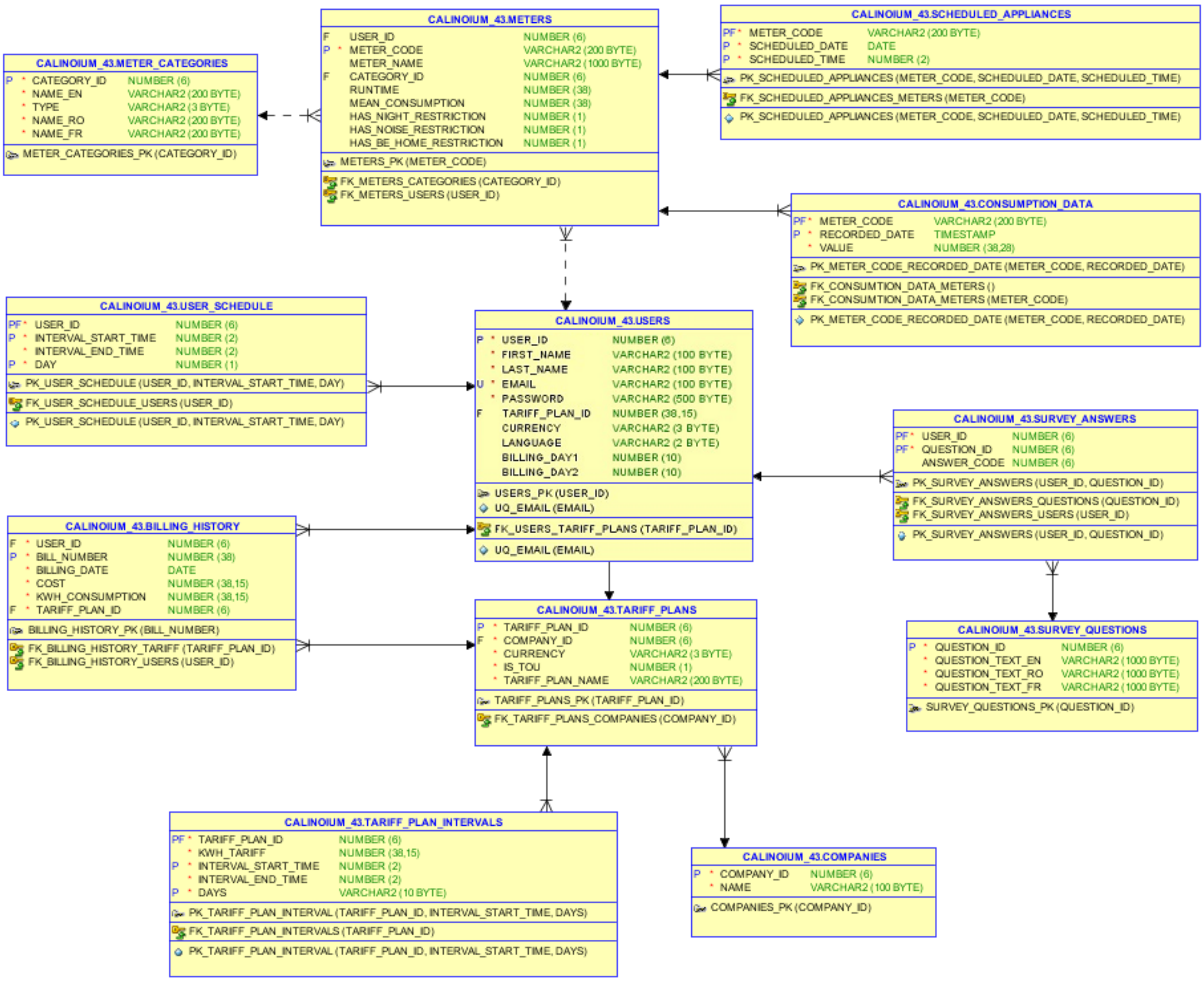


Fig. 1. Logical Schema of the Database

Reaching as many consumers as possible is one of the app's main objectives. Therefore, an Android mobile app provides an excellent platform, being available to most of the population that uses a cellular phone. Furthermore, a mobile app is the most accessible, requiring the least amount of effort and technical skill to use.
Mobile user interfaces in Android are represented by activities and dialogs. Following the description of the mobile application's functionalities and the data

flow, Watt Manager's screens are shown schematically below, in Figure 2. Links between interfaces show paths that can be accessed by the users within each screen.

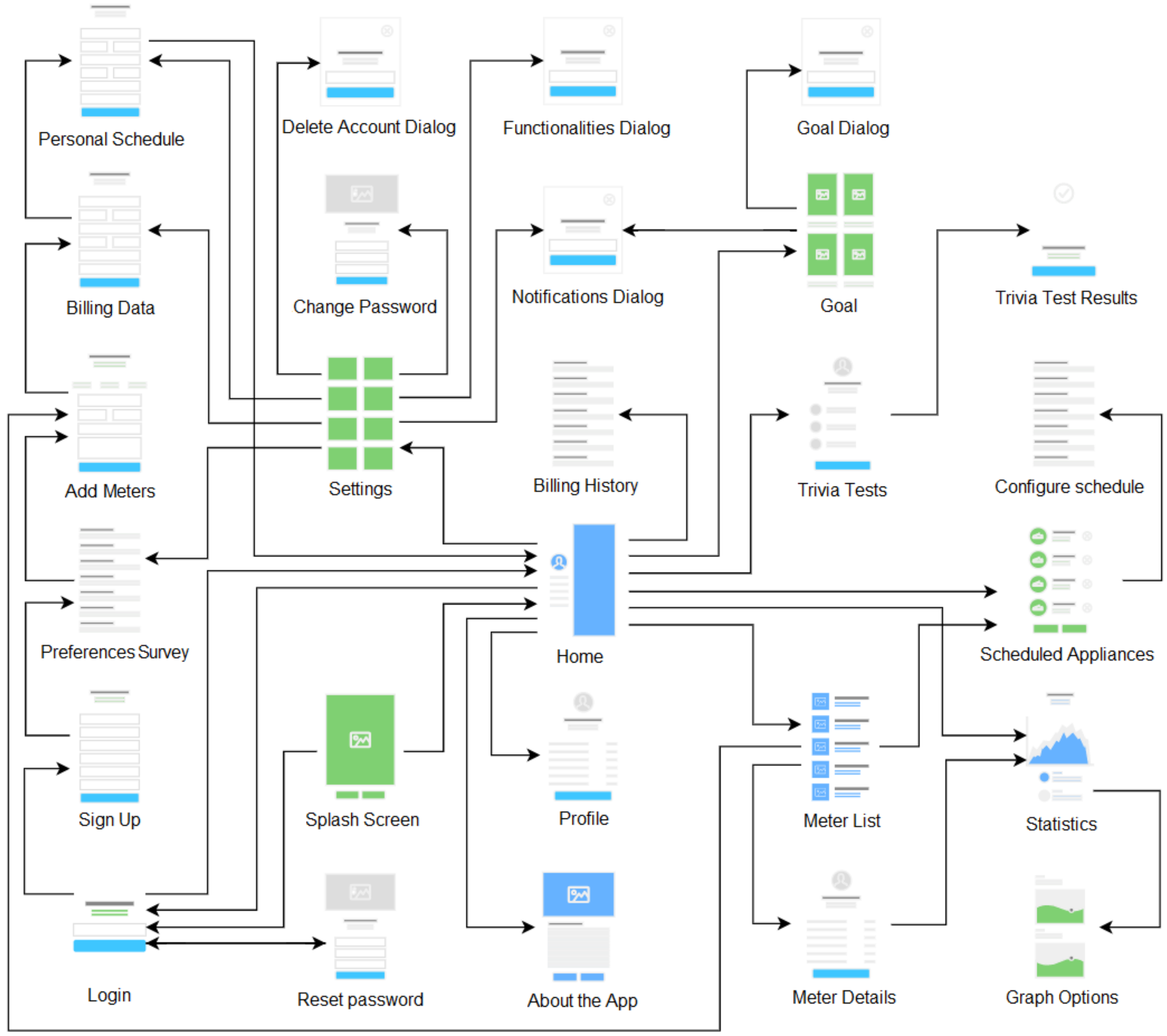


Fig. 2. User Interfaces Diagram

4 Software Technologies

Developing the application involves creating three software components: an Oracle relational database, a Flask REST server, and an Android mobile app.

JSON (JavaScript Object Notation) is a text format that uses key-value pairs. This format is used to transfer messages between the server and the app.

Extensible Markup Language, or **XML**, encodes documents using tags marked with symbols "<>", which describe the transmitted data. They are used in the mobile app as compiled resources to define the visual elements that make up the user interface, character strings, menus, colors, styles, and sizes.

Representational State Transfer (**REST**) is a client-server software architecture that acts as an intermediate level between the application and the database. A **REST-API** (Application Program Interface) separates the user interface from the data storage.

Isolating these components has several advantages. First, the application becomes scalable. Also, connecting to the database in a controlled way makes the process more secure. At the same time, it ensures portability, and the possibility of extension without constraints regarding the connection to the database. Moreover, it allows the database or mobile application to be changed, or to extend the system by adding an iOS application, a Web platform or a module for power suppliers, without requiring major changes to the other components. Finally, since the architecture implements a stateless protocol (**HTTP**), it is not necessary to load the server with data prior to each request because the information is sent in such a way that it can be understood without taking into account previous messages.

The mobile application places requests to the server through a uniform resource identifier (**URI**), the web address the server is running. The **GET** method is used to retrieve data, the **POST** and **PUT** methods are used to store the transmitted information into the database, and the **DELETE** method erases information from the database.

Python is an intuitive, high-level, dynamic, general-purpose programming language. The software principles on which Python was developed are synthesized in a collection of 19 expressions, "The Zen of Python", by one of the developers of language, Tim Peters. Among them, are: “Beautiful is better than ugly”, “Explicit is better than implicit”, “Simple is better than complex”, “Flat is better than nested”, “Sparse is better than dense”, “Readability counts”, “In the face of ambiguity, refuse the temptation to guess” [7].

The Python standard library is very large compared to other programming languages. Python Package Index, the official repertory, offers over 130,000 libraries. They facilitate both software development and rapid data processing in the exact sciences. Since most libraries are open-source, development in Python offers the programmer flexibility.

There are a number of Python development environments, such as *PyCharm, Jupyter Notebook, or Amazon Web Service Cloud9*.

Amazon Web Service Cloud9 is an integrated development environment based on Amazon Cloud. The advantage offered by Cloud9 is providing the URI required for HTTP calls to the REST server. Normally, non-cloud-based development environments either allow the creation of a local server or require the configuration of an external server. Since the application architecture is client-server, it is essential that the remote server exists to allow the interaction between the REST-API and the mobile devices running the application. This platform provides the programmer with an external IP that makes this possible.

Flask is a Python micro-framework designed for server development, following the REST paradigm for system interaction. The Watt Manager server was built with Flask. The messages transferred between the app and the server are parsed by the server with the **JSON** library. To retrieve data from the Oracle database, the connection is established with the help of the **cx_Oracle** library, and the queries use its *Cursor* object. User passwords were secured by encoding them in the database as *hash* values using the *Security* module in the **Werkzeug** library. The libraries used for data processing include **NumPy** and **Pandas**. NumPy is the fundamental package for mathematical calculations, with many predefined mathematical functions. Pandas provides data structures (such as *Series* and *DataFrame*) and functions for manipulating data.

Java is an object-oriented programming language, created to facilitate the development of distributed applications, which can be executed on heterogeneous networks [8]. Java is also used to develop mobile applications for devices that use the Android operating system.

Android Studio is the official Android app development environment, created by Google. Watt Manager was developed in Android Studio for a minimum level API 21, the equivalent of Android 5.0 (Lollipop), and a target level API 28 (Android 9). Thus, approximately 89.3% of existing Android devices can run the application [9]. The app only requires an Internet permission, being based on the client-server architecture which requires online communication.

The **Gson** library is implemented by Google in order to allow the conversion of Java objects or of user-defined serializable objects to JSON. Also, Gson can transform JSON character strings into objects. This quickly generates the POST and PUT requests, eliminating the need for manual definition. **OkHttp** is an HTTP client which facilitates the creation and execution of the requirements in Android towards the remote server. **MPAndroidChart** is an Android library used to build the charts in the Statistics screen, offering the ability to draw and personalize complex graphs with ease.

Relational databases are built on relational set theory. They use attribute integrity restrictions to control the data's accuracy, and associations between entities to ensure consistency. The tables are normalized, giving them a structured character, with minimal and controlled redundancy. Moreover, they support relational algebra operations, such as: union, intersection, difference, Cartesian product. In addition, they also support specific SQL operators. The advantage of using an SQL database is the mathematical, intuitive character of its design, construction, and query.

Structured Query Language, or **SQL**, is a programming language used for interrogating and manipulating relational databases. Procedural Language/Structured Query Language, or **PL/SQL**, is an Oracle extension of the SQL language. It contains modular code blocks stored directly in the Oracle database, enabling data control, manipulation and validation. SQL is used when creating, modifying, and querying the app's database, while PL/SQL is used to create triggers and sequences that generate indices automatically.

Oracle SQL Developer is a development environment used to build and to manage Oracle relational databases, implementing SQL and PL/SQL. It allows concurrent access and ensures data persistence using a remote database server.

5 Methodology

The shifting optimization algorithm represents the most complex processing sequence in the app's server. It receives the scheduling date and a list of devices to be scheduled, together with the initial hours proposed by the user as input values.

The database stores the restrictions imposed for the algorithm. The first type of restriction is given by each device's category. If the dryer and the washing machine are programmed on the same day, for example, the dryer should operate immediately after the washer, being scheduled accordingly. A device can have three optional restrictions: it can't be planned at night, it can't be planned during quiet hours, and it can't be planned in the absence of the user (they must be home during the appliance's runtime). At signup, users who benefit from ToU rates are invited to provide their approximate weekly schedule. This is the point of reference for the third restriction described above.

The algorithm's goal is to flatten the consumption curve by minimizing the peak usage, considering the constraints outlined above. Starting with the hours proposed by the user for each appliance, the algorithm follows an iterative process to gradually reduce maximum consumption, until it can no longer be improved.

The algorithm is adapted from the implementations presented in [5] and [6], with added user-imposed constraints for each device. Also, the way dependent and interruptible devices are handled is different. To manage dependent devices, they are grouped together, forming a single device with a summed runtime. By contrast, interruptible devices are divided into several uninterruptible devices, one for each hour of the initial device's runtime.

The algorithm is described with the help of the logical schema presented in Figure 3.

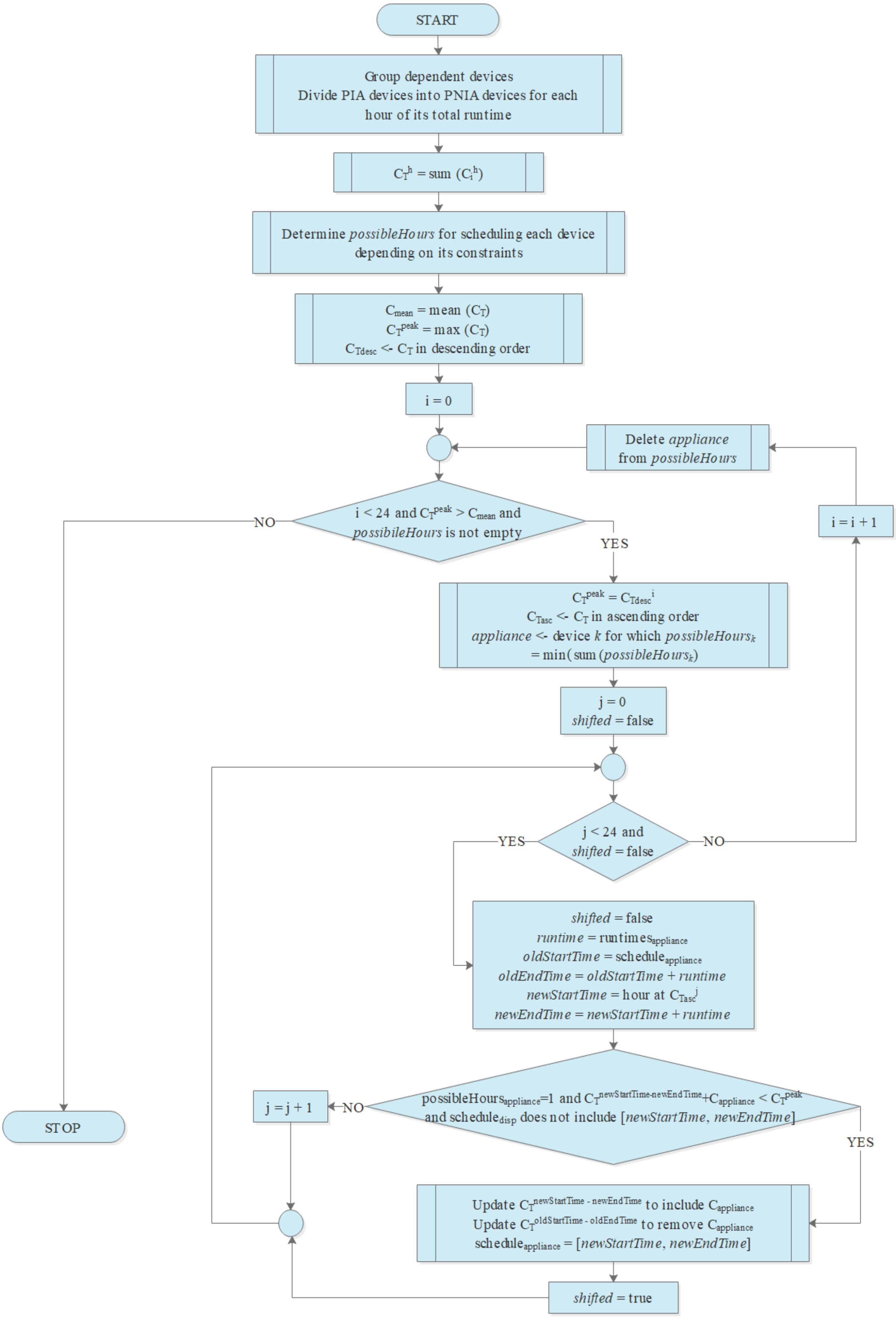


Fig. 3. Logical Schema of the Shifting Optimization Algorithm

C_T	Total consumption array for each hour

possibleHours	Boolean matrix with indices representing scheduled devices and hours representing columns, indicating possible scheduling times for device i at hour j, depending on its constraints
C_{mean}	The mean of the C_T total consumption array
C_T^{peak}	Peak consumption, calculated as the maximum of C_T
C_{Tdesc}	Array C_T in descending order
C_{Tasc}	Array C_T in ascending order
appliance	The appliance with the least number of possible scheduling hours, calculated by obtaining the index of the row with the minimum sum in the *possibleHours* array
shifted	Boolean variable signaling whether the current device has been shifted during the current iteration
runtimes	Array that stores the approximate runtime for each device scheduled
oldStartTime	Scheduled time for the device before it is shifted
oldEndTime	Time until the device runs before it is shifted, obtained by adding its runtime to *oldStartTime*
newStartTime	Scheduled time for the device after it is shifted
newEndTime	Time until the device runs after it is shifted, obtained by adding its runtime to *newStartTime*

6 Application Interface Depiction

Using the Watt Manager mobile application begins at signup, with the registration of a customer who owns smart meters that record consumption data. First, they introduce their personal data. Then, they complete the preference survey, add their meters by code, complete the billing form, and add their personal schedule.

Adding a meter requires a user-provided code. If the code exists in the database, it is validated and associated with the user, along with the name and category given. If the selected category represents programmable devices, the user also enters its runtime and its schedule restrictions in order to ensure the correct operation of the planning algorithm in the future.

Then, after adding their meters, users continue to the billing data form. If the selected tariff plan is a ToU plan, they introduce their personal weekly schedule. After completing the process, the client accesses the Home screen of the application. This screen also differs depending on the tariff type.

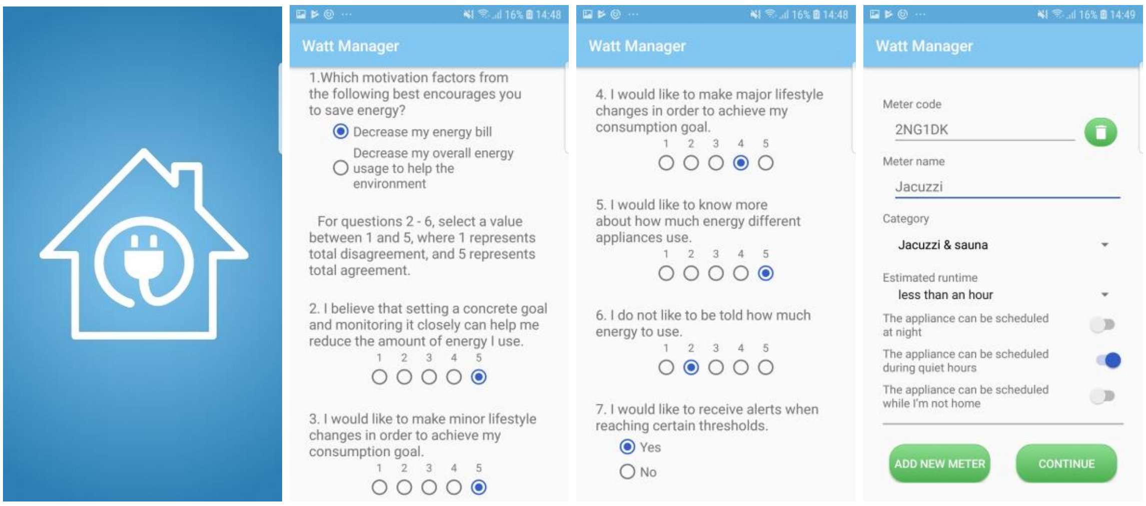


Fig. 4. Loading Screen, Survey, and Adding Meters

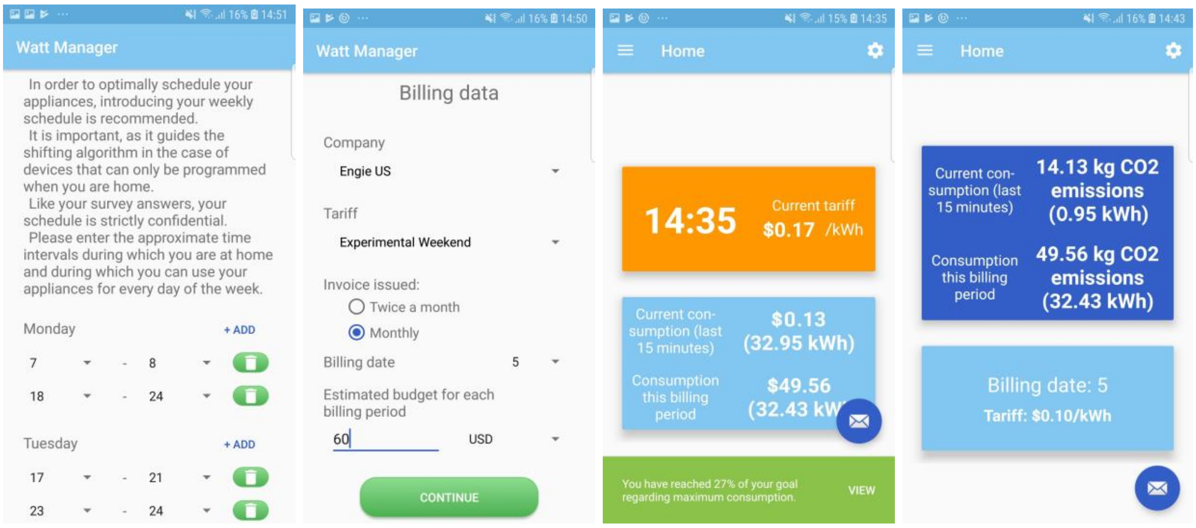


Fig. 5. Personal Schedule, Billing Data, and Home Screens

From the app's menu, users can access all of the app's functionalities.

The main screen contains two boxes and a Floating Action Button. The button shows the user's goal progress, and pressing it opens the goal screen. If the user benefits from a ToU tariff plan, the first box displays the current hourly rate. It is colored green, yellow, orange, or red, depending on the current cost of electricity relative to the minimum, maximum, and average tariff in the customer's plan, visually representing the current tariff. The second box shows current and total consumption for the time unit selected. If the user does not have a ToU tariff, the first box indicates total usage, and the second box shows billing details and the constant rate per kWh.

Selecting the box showing the consumption sends the user to the Statistics screen. By default, it displays separate consumption data for each meter so that the customer can better understand their consumption habits. If the user wants to generate another graph, they provide the measurement unit, the time unit, and the type of chart desired. Depending on this information, the user chooses the type of consumption (total, aggregated by category, aggregated by tariff, or disaggregated) and the app generates a chart.

Selecting the box that displays the tariff leads to the user's profile, including a more detailed view of the tariff plan.

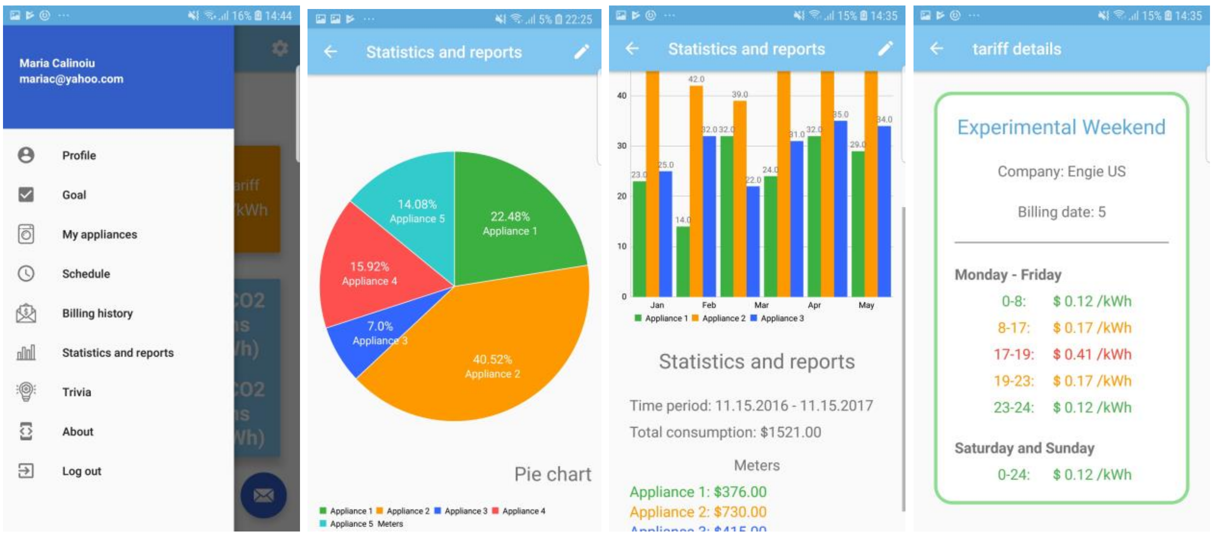


Figure 6 – App Menu, Statistics, and Tariff Details

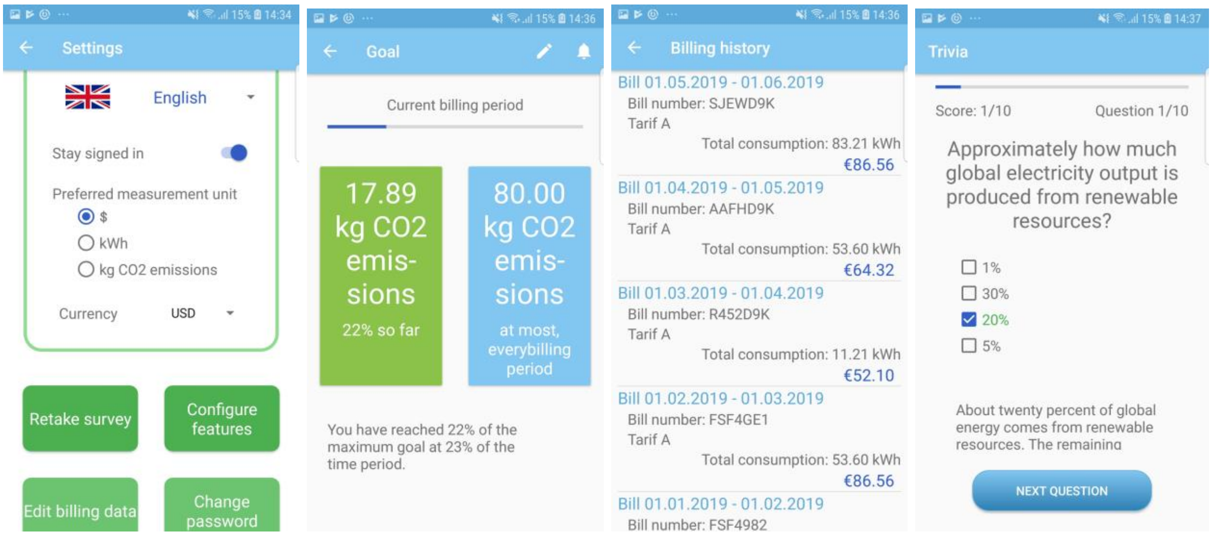


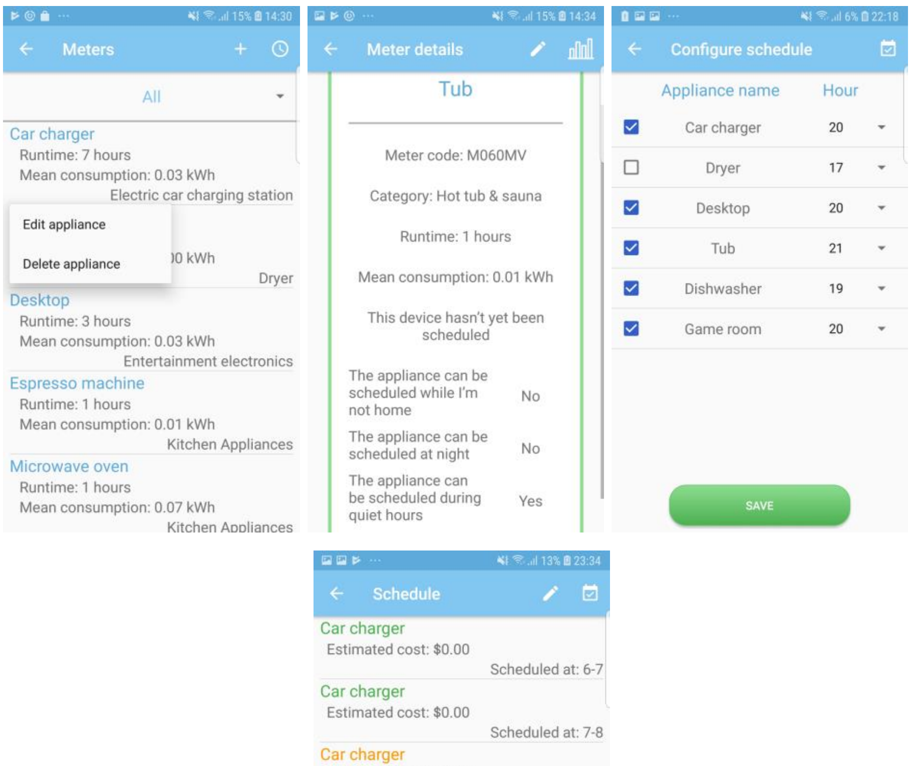
Fig. 7. Settings, Goals, Billing History, and Trivia

App settings allow the user to edit the data recorded at signup, except for their personal data. The user can also change their password and delete their account, enable or disable features, and configure notifications.

The Goals screen shows the current progress during the period for which the target is set. Like the Home Page box, the progress box is colored from green (0% of the maximum usage reached), to red, (exceeding the maximum usage set). To set a goal, the user accesses the dialog from the header menu. There, they can also enable or disable goal notifications.

Trivia tests contain ten questions regarding electric energy, aimed to develop consumer knowledge and to raise awareness regarding the impact of daily usage on the environment. After selecting and confirming an answer, the correct response is indicated, along with an explanation.

The appliance list displays the user's meters. Clicking a device shows its details, and long-clicking it displays the context menu that allows editing or deleting it. The scheduling feature can be accessed from the Appliances header menu, leading to a screen where the user can select which devices to program, the estimated scheduling time, and the date. After running the algorithm on the server, the app returns its result in the scheduling screen. There, the names of the planned devices are colored according to the tariff value at their programmed time.



7 Conclusions

The objective of the paper was to analyze the best way to implement an application for monitoring and controlling electricity consumption, regarding the functionalities included and the technical aspects involved. The positive impact on the environment produced by decreasing usage and adapting consumer behavior to avoid overloading the distribution network provided the motivation for choosing this subject.

The proposed IT solution facilitates consumption surveillance by providing the most relevant feedback to each user. The profile established on the basis of a questionnaire completed at signup captures the user's motivational factors and preferences. This determines the content of the application, in order to determine users to reduce their consumption or to optimize it. The application follows a three-tier architecture: an SQL relational database, an intermediate REST server developed in Python Flask, and an Android mobile app developed in Java.

The existing application can be improved by building the necessary hardware components for measuring consumer data. In addition, considering multiple sensors can provide additional details on consumer behavior. Thus, recommendations become even more personalized. Also, adding a Watt Manager version compatible with iOS devices or other operating systems would be an important step to cover a larger proportion of consumers.

The system can be expanded by developing a Web platform, especially for energy providers, where they can visualize the users' consumption profiles, as well as changes in behavior based on pricing, season, or other stimuli tested. Suppliers can also obtain estimates based on consumer data to manage their business effectively.

European policies in the energy field, technological developments, and the need to control electric energy usage highlight the importance of such an application.

8 Acknowledgments

This paper presents the scientific results of the project "Intelligent system for trading on wholesale electricity market" (SMARTRADE), co-financed by the European Regional Development Fund (ERDF), through the Competitiveness Operational Programme (COP) 2014-2020, priority axis 1 – Research, technological development and innovation (RD&I) to support economic competitiveness and business development, Action 1.1.4 - Attracting high-level personnel from abroad in order to enhance the RD capacity, contract ID P_37_418, no. 62/05.09.2016, beneficiary: The Bucharest University of Economic Studies.

References

[1] European Comission, "Eurostat," European Comission, 17 08 2018. [Online]. Available: https://ec.europa.eu/eurostat/web/energy/data/main-tables. [Accessed 17 06 2019].

[2] R. S. Brewer, Fostering sustained energy behavior change and increasing energy literacy in a student housing energy challenge, Manoa: ProQuest LLC, 2013.

[3] B. Karlin and J. F. Zinger, "The Effects of Feedback on Energy Conservation: A Meta-Analysis," *Psychological Bulletin*, vol. 141, no. 6, pp. 1205-1227, 2015.

[4] S. Darby, "Smart metering: what potential for householder engagement?," *Building Research & Information*, vol. 38, no. 5, pp. 442-457, 2010.

[5] S. V. Oprea, A. Bâra, A. I. Uță, A. Pîrjan and G. Căruțașu, "Analyses of Distributed Generation and Storage Effect on the Electricity Consumption Curve in the Smart Grid Context," *Sustainability*, vol. 10, pp. 2264-2289, 2018.

[6] S. V. Oprea, A. Bâra and G. Ifrim, "Flattening the electricity consumption peak and reducing the electricity payment for residential consumers in the context of smart grid by means of shifting optimization algorithm," *Computers & Industrial Engineering,* vol. 122, pp. 125-139, 2018.

[7] T. Peters, "PEP 20 -- The Zen of Python," 19 August 2004. [Online]. Available: https://www.python.org/dev/peps/pep-0020/.

[8] Sun Microsystems, Inc., "JavaSoft Ships Java 1.0," Sun Microsystems, Inc., 23 01 1996. [Online]. Available: https://tech-insider.org/java/research/1996/0123.html. [Accessed 18 05 2019].

[9] Google, Inc., "Distribution dashboard," Google, Inc., 07 05 2019. [Online]. Available: https://developer.android.com/about/dashboards. [Accessed 18 05 2019].

Maria Irène CĂLINOIU (b. August 9, 1997) is a graduate of the Faculty of Economic Cybernetics, Statistics, and Informatics at the Bucharest University of Economic Studies, under the tuition of PhD. Lecturer Simona Vasilica Oprea. She graduated “Tudor Vianu” National High School of Computer Science in Bucharest in 2016. Her scientific fields of interest include Programming, Databases, Data Science, Natural Language Processing, and Machine Learning.

Simona-Vasilica OPREA (b. July 14, 1978) received the MSc degree through the Infrastructure Management Program from Yokohama National University, Japan, in 2007, the first Ph.D. degree in Power System Engineering from the Bucharest Polytechnic University in 2009, and the second Ph.D. degree in Economic Informatics from the Bucharest University of Economic Studies in 2017. She is involved in several national and H2020 European research projects as member or project manager. She is currently project director for a H2020 project entitled Multi-layer aggregator solutions to facilitate optimum demand response and grid flexibility (acronym SMART-MLA).