UGent

Algorithms and Data Structures Project

ENERGY CONSERVATION SYSTEM'S DESIGN AND IMPLEMENTATION

MARIA JOSE ALEY JIMENEZ AND VICENTE SOTO JARPA

Content

Introduction

This project involves the creation of an application that serves as a software that can provide services for students and landlords in the administration of energy consumption, the appliances, and the actions made to directly impact on energy consumption. The main goal of the project was to create and develop a system that can fulfill all the requirements established by the “users” as it would happen in real life.

Due to our inexperience in the matter, a lot of the time was used doing research about multiple frameworks and ways to approach the problem. In short terms, we determined to use Django as our main framework, which lend us the chance to implement all our specifications using the Python language and its packages. As and IDE, we used the community version of Pycharm and Visualstudio. For the database, we decided to use PostgreSQL and PgAdmin to run it. Django offered a simple way to connect to the database, together with the option of working in modules with different applications that the project should consider, giving us a robust framework to implement our program and necessities.

Summary of project timesheet

The project was designed and implement in a timeframe of 8 weeks. The students worked to accomplish the different activities, from the research to the contribution of the system's design and implementation.

Time Sheet																
Activities/ Contributions	**Week 1**		**Week 2**		**Week 3**		**Week 4**		**Week 5**		**Week 6**		**Week 7**		**Week 8**	
Research and design	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
User Models	X	X									X	X				
User Views			X	X												
User forms			X	X												
User Templates					X	X	X	X								
Room Model							X	X	X	X					X	X
Room View									X	X						
Room forms									X	X						
Room Templates											X	X				
Appliances Model													X	X		
Appliances Views													X	X		
Appliances forms															X	X
Appliances Template															X	X
Reports views & templates															X	X

Maria	X
Vicente	X

System design and implementation

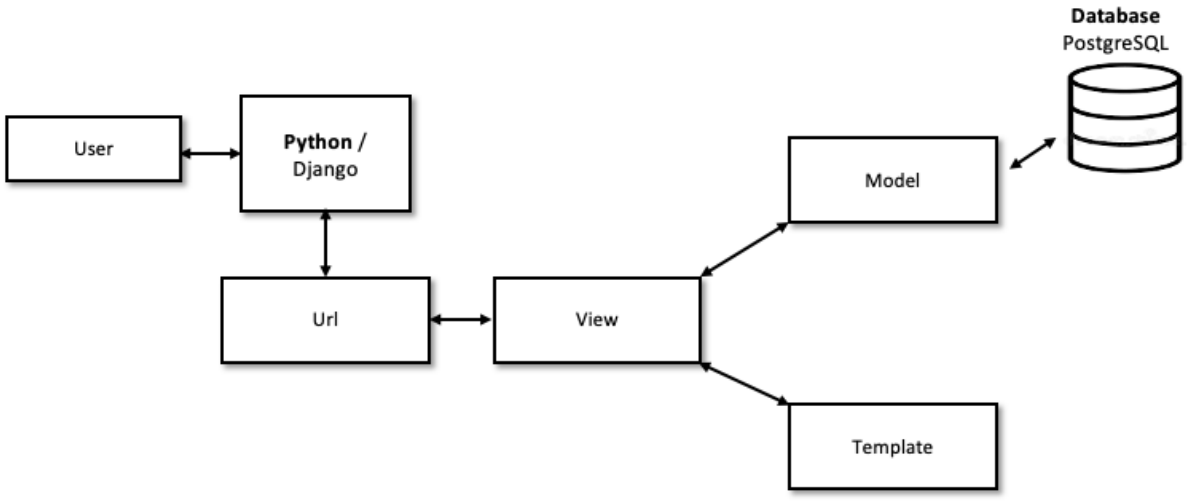
Software architecture

The software architecture of the system is based on the MVT (Model-View-Template) architecture.

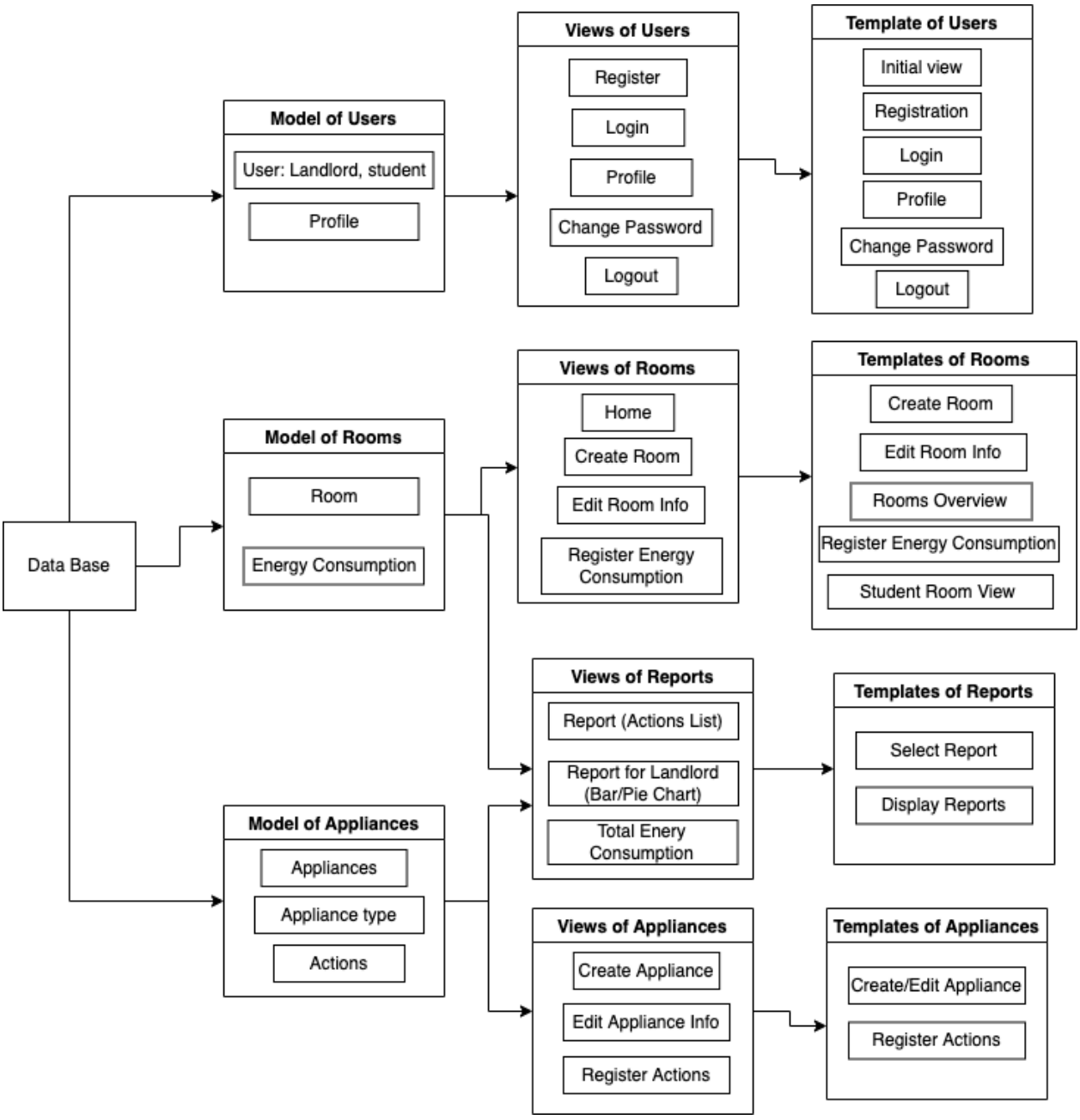
The model acts as the interface for the data. It is the data structure behind the application and is represented by a database, which is PostgreSQL in this case. The models created for the system are for users, rooms, and appliances.

The view acts as the controller of the requests and returns a web response by rendering the HTML/CSS of the templates files into what is displayed for the user in the browser.

The template handles and generates the HTML webpage, the layout of the website that are visible to the end-user. For each part of the project, different templates directories were created.



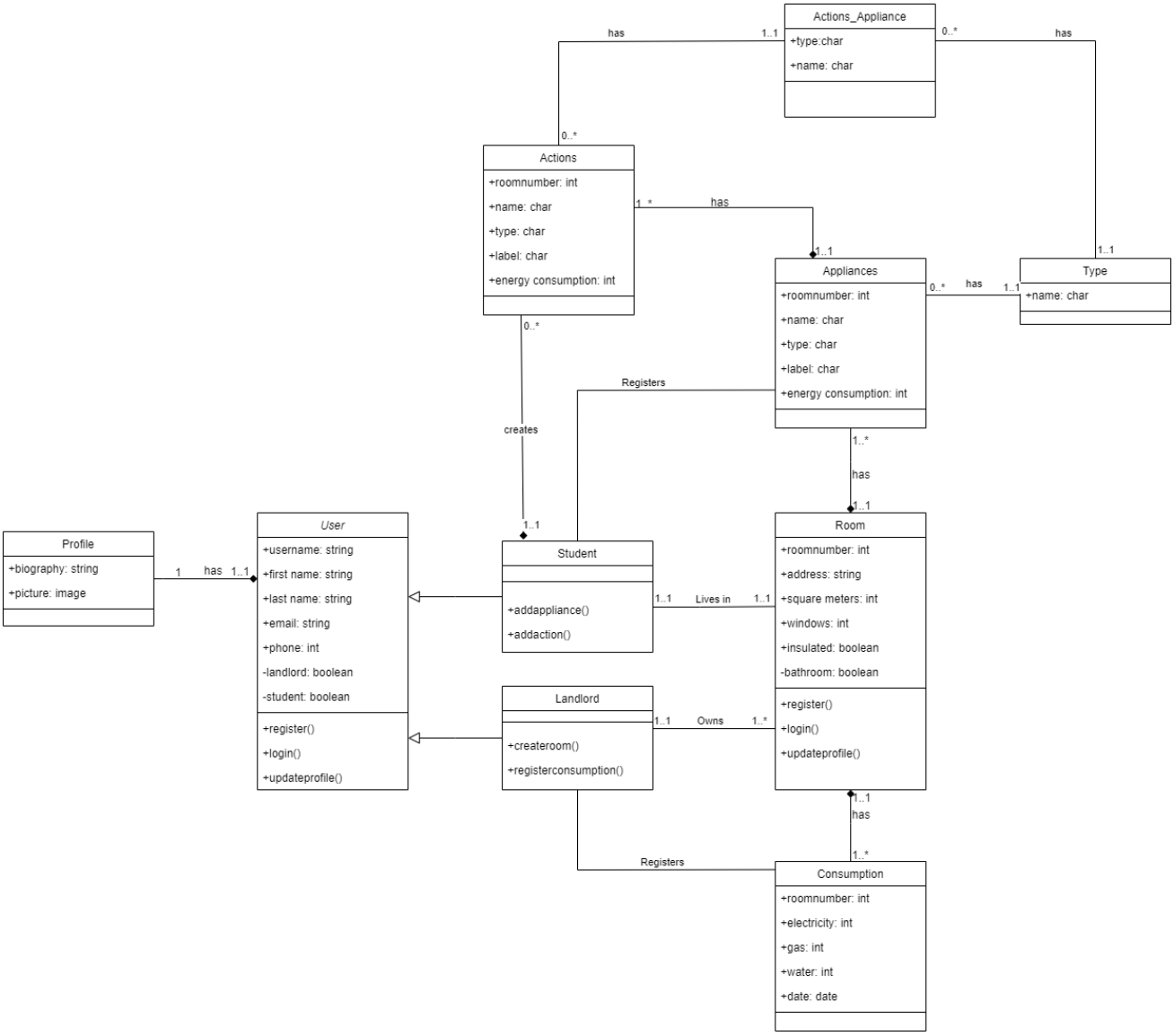
The software architecture of the project can be graphically represented as follows:



UML class diagram for every module

In this section we present the UML Class Diagram. In short terms, a class diagram is a static structured diagram that is intended to represent a system by representing the objects in it, with it classes, attributes and methods, along the relationship between them. This can be used as a guideline for new users to understand the system or even, if the model is correctly built, programmers could use it to build the code that actually make the program work. This can be also

used as the data model, but in this case, we built the entity-relationship model of the database that betters represent the database model.



As a simple description for it, every class has a diagram or “box” that consist of 3 “sub-boxes” In the first one, we write the name of the class. In the second or middle box, we write the attribute of the classes and in the third or bottom box we write its methods, which are the operations the class can execute.

As an example, we will explain the class diagram of the users. In this case the name of the class is “User”, and its attributes are username, first name, last name, email, phone, if it is a landlord or not, and if it is a student or not. We can also notice that along every attribute, we can find the type of variable that the attribute is. In this case the first and last names are char or string variables. The phone itself it is a number or int variable. In the last box we find the operations that a user can perform, in this case, all users can register, login and update their profile. One can notice at the diagram, to the left of the User class, that there are two sub-classes. This is represented by a relationship line with a white-headed arrow. This implies that the sub-classes “student” and “landlord” has the same attributes and methods that the “mother class” but they may also have new attributes. In this case, those classes do not have any new attributes, but they have new methods. For Students, they can add appliances for their room and add actions they have made to save energy. In the case of the Landlords, they can register new rooms and register the water, electricity and gas consumption for every room.

Also, we can find some numbers along every relationship in the class diagram. This numbers notate the cardinality between the relationships of the classes. For example, a student can perform zero or many actions, which is notated as “(0..*)”, but an action can be performed by just one student, so it is notated as “(1..1)”.

We just went over a quick overview of how to read a class diagram but we won’t describe the whole diagram since it is much more self-explanatory and that is itself the purpose of the diagram.

Implementation

Energy Conservation is a platform that allows the user to register and analyze the electricity usage in student rooms.

The business entities that are manage by the implementation are:

- Users: The user is defined as the person who has access to the platform and can be define either as a landlord or as a student. The information of the user needed requested in the registration are, username, name, last name, email, phone number, and to choose the status of landlord of student which will define the functionalities in the platform.

Functionalities	
User Landlord	Add and edit student rooms and link the room to a specific student
	Register the energy consumption of the room every month
	Generate Reports of the montlhy energy consumption of the Rooms and the sum of the energy usage of each room month by month
User Student	Appliances administration is performed by students, adding and edditing the appliances characteristics
	Register the energy conservation actions perfomed per appliance on an specific date
	Generate Reports of the energy conservation actions and display the most use actions through the month

- Rooms: The rooms are defined as the location in which the registration and analysis of the electricity usage will be made. The information of the room required are the address, room number, square meters, quantity of windows, and choosing on the bathroom and insulation status. The registration of a new room is a function that can be performed by a user defined as a landlord and it is necessary to create the link with the user defined as student and the date of registration in order to start monitoring the energy consumption.

- Appliances: The appliances in the room must be registered and specific characteristics are recorded like the energy consumption per year, the class label following the European energy label, the date of registration and a custom name. The appliances can be registered by the users defined as students and the appliances should be linked to a limited type of appliance.

- Energy conservation actions: The users defined as students can register energy conservation actions that they have performed through a specific date, these actions are linked to a specific appliance and the appliance type defines the different energy conservation action that can be registered.

- Energy Consumption: To keep track of the monthly energy consumption of each one of the rooms that have been register, the user defined as landlord must register electricity, water, and gas consumption from the room that the user has previously created.

- Reports: Reports are generated to analyze the monthly energy consumption that have been register by the landlords of the rooms and the actions of energy conservation that have been performed by the students.

The GitHub repository can be found at: github.com/maariaaley/energyconservation

The framework and libraries used to develop the application to support the functional requirements of the project are:

Tools	Functionality
Django	Python framework for creation of web applications
PostgreSQL	Object-relational database
Bootstrap	CSS framework for the design of templates
Crispy Forms	Application to manage Django forms
Slick Reports	Report engine for creation and display of analytics reports

Algorithmic support

In this section we describe the algorithmic support that can be achieved considering all the functions our application includes. Mainly, this means that based on our system implementation, we could derive certain new functions that could help the user take decisions or enhance their energy efficiency. For example, they could decide to implement new energy conservation actions based in what other users commonly use.

A recommendation system could be implemented using just simple statistics, for example, when a student is wondering what kind of actions, he could take to improve his energy savings, he could look at general reports of common actions taken by other students to save energy, so he could imitate his partners in case he runs out of ideas. Also, landlords could decide to send recommendations for the students based on what other students usually do for similar appliances. An example for this could be disconnecting the tv or the notebook charger every night, since landlords has access to the multiple rooms of multiple students in a lot of cases.

A more advance implementation that could be developed in the future, is to make use of more sophisticated statistics or even machine learning. When the system has enough data, this means, after a considerable amount of time being used by both students and landlords, we will have enough amount of data to take some decisions based on what data gives us in information. For example, we could try to create a regression model between the electric energy consumption and the actions taken by any student. In this way, the model with show us the actions with greater coefficients (negative coefficients), this means, that that kind of actions further derive in a reduced energy consumption. We could also add the appliances in the models as binary variables, as if the room has the appliance, then we would hypothesize that it will have a significant positive coefficient, which means that it will increase the energy consumption. In this case, students will try to take actions related to that appliance or cut down the appliance usage as much as possible.

Something similar could be implemented for water usage, where we take water saving actions and try to predict their influence in the variability of the water consumption. Also, something similar could be implemented for gas usage.

Conclusions and limitations

Limitations and Improvement possibilities

In this section we will go over all the limitations that we encounter while developing the project. It is a bit obvious that our main limitation relies on our knowledge and capabilities of developing a web application. Since we were new into a working framework suck as Django, for developing web applications, we had to invest too much time in researching and learning how to work with suck a framework. The same counts for PostgreSQL and the connection to the database system. Due to this main factor, a lot of improvement possibilities remain latent and could be implemented in the future with a bigger team or more time to develop it. Also is it worth mentioning that we discuss only some contention points since there are too many and we could spend the whole evening discussing them.

As mentioned in the project instructions, a great reflection point could be the fact that students could introduce their own actions or appliances to the system, so we can cover a greater range of opportunities to improve. The problem with this is that often, when you want to do some statistical analysis like work with aggregations, machine learning, etc. you need the names to be the same, so that the system recognizes them as the same action. This problem is presented very often is most databases and data engineers need to spend a lot of time cleaning the data in order to be able to work with it. A simple way of leading with this problem is standardizing the options so that the users can only pick out of a list of limited options so there won't be room for any "typing" mistake. Another approach to give a solution for this problem, is that student can propose new actions or appliances to a system manager or administrator, so that he can add these actions or appliances to the system without losing any standardization. Also, we don't want the data to get "dirty", this mean we don't want to add actions or appliances that no one will use, since this ones will be consider as outliers and will not generate relevant information.

Another contention point is it worth mentioning, is that technology advanced in a fast pace, so some actions that could derive in reduced in electric consume for some students do not do it for others. For example, modern computers and cellphones do not generally have "vampire" consumption of energy, this is because they have a built-in protective mechanism to prolong the battery lifespan. This means that when a computes battery is full and out of use, the computer will stop draining energy until the battery drops to a lower level (around 95%), so it will charge up back again. This can make students that do not disconnect their consumer have no extra electricity consumption that students who does.

One of the biggest contention points is that most of the electric consumption comes from certain appliances that have an anergy consumption much greater than other. For example, fridges and electric stoves consume way more energy than mobile phones or led bulbs, so actions for saving energy like turning off the lights could be almost negligible in comparison of having an electric stove on for a few minutes.

General conclusions

In general conclusion, we could achieve great part of the project, handing in a system that fulfills almost all of the requirements with an interactive graphic interface, easy to learn and use for both students and landlords. This system is fully connected and with the help of Django we could work with login and user authentication, this also provided us with the tools to make the system safer for the users, since out program also works with encrypted passwords, so not even us or IT managers, with access to the database, could see the passwords. It is also worth mentioning that although there is still room for improvement, this is always the case in most of the software or applications in the world, since all of them are always releasing new versions, including new functionalities and bug fixing. The overall result accomplished gives a very strong base to work at, since it is modular, it gives a robust application that can adapt to a lot of situations and changes that could be required in the future. All of its functionalities can be modified in accordance of the requirements of the user and adapt to bigger challenges.