# Video demonstration

<https://youtu.be/az6jo-7xni4>

# Technical report

I have a Smart Home console application which enables the user to input data through the use of the console, and that data is then processed and outputted to display information about smart plugs.

There are 6 parts to my project a Main which is called the dashboard, a Console help class which controls statics methods to allow for easier outputting and inputting, a Smart home object which controls all 3 sub objects, which includes the smart plug, room and device objects.

All data is stored separately away from the user inside these 3 main objects. A small amount of data is stored in the other to represent index values to be used for loops and holding places in the array.

## Main

The main program is created inside dashboard which holds the main method. This method is used to initialize the Smart Home object which stores all smart plugs, rooms and devices, and the Console helper object which stores useful static helper methods as well as the main code for the dashboard.

After initializing these the user is asked for the number of plugs and the number of rooms which are sent to these initialized objects.

A populate method is then called which enables the user to fill all the room names, and a second populate method is then called to allow the user to set the data in for the plugs.

After this a while loop is used to display a desktop. It calls the Console helper displayDashboard method and the displayOptions method, this while loop continues until a return value of 1 is sent back by the console helper object.

## Console Helper

The console helper once called allows the user to input a integer which controls the option type they would like to use. This data is used to control a switch statement, and dependant on the int given a method is called.

These methods are passed the object smart home and are stored in the console helper as well. They allow for a second input from the user for more specific choices, and control the data entered for example they allow the user the enter a new name for a room. Once their purpose has been fulfilled, they will call Smart home object which controls all plug data.

There are also an additional 3 methods inside the console helper to control the adding of new data for new objects. They allow the user to input data which is then sent to the Smart home.

## Smart Home object

The smart home object is the main object which controls almost all operations on data. Through the use of a constructor, it initializes all arrays of objects for the plugs, rooms and devices which all their data to be inside.

It contains 4 methods to output the size of the arrays. It also has the methods to create the initial objects for the plugs, rooms and devices (created by the program on run not by user.)

These allow for passed data by the console helper.

There are then 4 methods to control the outputting of the objects data this is done through the use of a loop and string which stores the toString method used for each object. And to get the correct device/room name the getters are used to get the ID stored inside each plug.

There are then 9 methods to control the options that allow operations, deletion and creation on/of the objects.

The first 4 methods allow the plug status to be updated. The first method allows for a state to be given and this is applied to all plugs. The second one allows for a state to be given and a room Id this updates all plugs in that room with the status given. The third allows for a ID to be given this toggles the current state of the plug with that specified ID. The fourth method is the same as the third, however it allows a state to be specified instead of the state being flipped.

The other 2 methods control the changing of devices and rooms. They allow for the ID of the plug to be passed and allow for the ID of the device/room to be passed. The sPlug[ID] is then required to use a setter to update the device/room to the new ID.

The last 3 methods control the adding of new plugs, device & rooms. They allow for a number of new items to be passed. This int is then added to the size method to produce a new array size used to initialize a new temp array this is then looped to allow the old arrays objects to be stored in the new temp array. The old array is then updated with the length of the temp array, and then all data is passed back to it through another loop.

## Smart Plug object

The Smart Plug object stores 4 pieces of data it stores the ID of the plug which is generated by a currentIndex value inside the smart home object. The status which is false until updated by the user. The deviceID and the roomID which are given by the user when a plug is populated, they are also updated by the user.

The smart plug object also contains getters and setters to allow the interaction with the data stored in the objects.

It also has a toString method which stores all data ready for it to be outputted, for the user.

## Room & Device objects

These two objects are very similar they store a constructor and have 2 pieces of data each. They store their ID and the name of the room/device. They also have getters and setters to allow these variables to be accessed from the smart home object, which is the object which controls all interactions with them.

They both also have toString methods which allow their data to be easily outputted to the user.