Julio Veganos e Hijos interface

Generated by Doxygen 1.9.1

1 Hierarchical Index **1**
1.1 Class Hierarchy . . . 1
2 Class Index **3**
2.1 Class List . . . 3
3 File Index **5**
3.1 File List . . . 5
4 Class Documentation **7**
4.1 AirConditionSensor Class Reference . . . 7
4.1.1 Detailed Description . . . 8
4.1.2 Constructor & Destructor Documentation . . . 8
4.1.2.1 AirConditionSensor() . . . 9
4.1.3 Member Function Documentation . . . 9
4.1.3.1 checkAirCondition() . . . 9
4.1.3.2 getAirCondition() . . . 9
4.1.3.3 showChoice() . . . 10
4.1.4 Member Data Documentation . . . 11
4.1.4.1 airCondition . . . 11
4.1.4.2 choice . . . 12
4.2 Camera Class Reference . . . 12
4.2.1 Detailed Description . . . 13
4.2.2 Constructor & Destructor Documentation . . . 13
4.2.2.1 Camera() . . . 13
4.2.3 Member Function Documentation . . . 13
4.2.3.1 turnOff() . . . 14
4.2.3.2 turnOn() . . . 14
4.2.4 Member Data Documentation . . . 15
4.2.4.1 state . . . 15
4.3 Dashboard Class Reference . . . 15
4.3.1 Detailed Description . . . 16
4.3.2 Constructor & Destructor Documentation . . . 16
4.3.2.1 Dashboard() . . . 16
4.3.3 Member Function Documentation . . . 16
4.3.3.1 getOption() . . . 16
4.3.3.2 showMenu() . . . 17
4.3.4 Member Data Documentation . . . 18
4.3.4.1 A . . . 18
4.3.4.2 D . . . 18
4.3.4.3 H . . . 19
4.3.4.4 L . . . 19
4.3.4.5 Li . . . 19

4.3.4.6 M . . . 19
4.3.4.7 option . . . 19
4.3.4.8 RC . . . 19
4.3.4.9 T . . . 20
4.3.4.10 TC . . . 20
4.4 DataBase Class Reference . . . 20
4.4.1 Detailed Description . . . 22
4.4.2 Constructor & Destructor Documentation . . . 22
4.4.2.1 DataBase() . . . 22
4.4.3 Member Function Documentation . . . 22
4.4.3.1 addUser() . . . 23
4.4.3.2 checkUser() . . . 23
4.4.3.3 deleteUser() . . . 24
4.4.3.4 saveFile() . . . 25
4.4.3.5 showAdminChoices() . . . 25
4.4.3.6 showUsers() . . . 26
4.4.4 Member Data Documentation . . . 27
4.4.4.1 choice . . . 27
4.4.4.2 dataBaseUser . . . 27
4.4.4.3 isAdmin . . . 27
4.4.4.4 it . . . 27
4.4.4.5 name . . . 28
4.4.4.6 NIF . . . 28
4.4.4.7 NIFStr . . . 28
4.4.4.8 type . . . 28
4.4.4.9 userDeleted . . . 28
4.4.4.10 userNow . . . 28
4.4.4.11 userNumber . . . 29
4.4.4.12 userNumberStr . . . 29
4.5 HumiditySensor Class Reference . . . 29
4.5.1 Detailed Description . . . 30
4.5.2 Constructor & Destructor Documentation . . . 30
4.5.2.1 HumiditySensor() . . . 31
4.5.3 Member Function Documentation . . . 31
4.5.3.1 checkHumidity() . . . 31
4.5.3.2 getHumidity() . . . 31
4.5.3.3 showChoice() . . . 32
4.5.4 Member Data Documentation . . . 33
4.5.4.1 choice . . . 33
4.5.4.2 humidity . . . 33
4.6 LigthLevelSensor Class Reference . . . 34
4.6.1 Detailed Description . . . 35

4.6.2 Constructor & Destructor Documentation 35
4.6.2.1 LigthLevelSensor() 36
4.6.3 Member Function Documentation 36
4.6.3.1 checkLigthLevel() 36
4.6.3.2 getLigthLevel() 36
4.6.3.3 showChoice() 37
4.6.4 Member Data Documentation 38
4.6.4.1 choice 38
4.6.4.2 ligthLevel 38
4.7 Login Class Reference 39
4.7.1 Detailed Description 40
4.7.2 Constructor & Destructor Documentation 40
4.7.2.1 Login() 40
4.7.3 Member Function Documentation 40
4.7.3.1 adminMenu() 40
4.7.3.2 checkLogin() 40
4.7.4 Member Data Documentation 41
4.7.4.1 NIF 41
4.7.4.2 NIFStr 41
4.7.4.3 userNow 41
4.7.4.4 userNowPtr 42
4.7.4.5 userNumber 42
4.7.4.6 userNumberStr 42
4.8 Microphone Class Reference 42
4.8.1 Detailed Description 43
4.8.2 Constructor & Destructor Documentation 43
4.8.2.1 Microphone() 43
4.8.3 Member Function Documentation 43
4.8.3.1 getSound() 43
4.8.3.2 showChoice() 44
4.8.3.3 turnOff() 44
4.8.3.4 turnOn() 45
4.8.4 Member Data Documentation 45
4.8.4.1 choice 45
4.8.4.2 sound 45
4.8.4.3 state 45
4.9 NIFException Class Reference 46
4.9.1 Detailed Description 47
4.9.2 Constructor & Destructor Documentation 47
4.9.2.1 NIFException() 47
4.10 RGBCamera Class Reference 47
4.10.1 Detailed Description 48

4.10.2 Constructor & Destructor Documentation 48
4.10.2.1 RGBCamera() 48
4.10.3 Member Function Documentation 49
4.10.3.1 getRGBImage() 49
4.10.3.2 showChoice() 49
4.10.4 Member Data Documentation 50
4.10.4.1 choice 50
4.11 Sensor Class Reference 51
4.11.1 Detailed Description 52
4.11.2 Constructor & Destructor Documentation 52
4.11.2.1 Sensor() 52
4.11.3 Member Function Documentation 52
4.11.3.1 turnOff() 52
4.11.3.2 turnOn() 53
4.11.4 Member Data Documentation 53
4.11.4.1 state 53
4.12 StringException Class Reference 54
4.12.1 Detailed Description 55
4.12.2 Constructor & Destructor Documentation 55
4.12.2.1 StringException() 55
4.13 TemperatureSensor Class Reference 55
4.13.1 Detailed Description 56
4.13.2 Constructor & Destructor Documentation 56
4.13.2.1 TemperatureSensor() 57
4.13.3 Member Function Documentation 57
4.13.3.1 checkTemperature() 57
4.13.3.2 getTemp() 57
4.13.3.3 showChoice() 58
4.13.4 Member Data Documentation 59
4.13.4.1 choice 59
4.13.4.2 temperature 60
4.14 ThermalCamera Class Reference 60
4.14.1 Detailed Description 61
4.14.2 Constructor & Destructor Documentation 61
4.14.2.1 ThermalCamera() 61
4.14.3 Member Function Documentation 62
4.14.3.1 getThermalImage() 62
4.14.3.2 showChoice() 62
4.14.4 Member Data Documentation 63
4.14.4.1 choice 63
4.15 TypeError Class Reference 64
4.15.1 Detailed Description 65

4.15.2 Constructor & Destructor Documentation 65
4.15.2.1 TypeError() 65
4.16 User Class Reference 65
4.16.1 Detailed Description 66
4.16.2 Constructor & Destructor Documentation 66
4.16.2.1 User() [1/2] 66
4.16.2.2 User() [2/2] 66
4.16.3 Member Function Documentation 67
4.16.3.1 checkAdmin() 67
4.16.3.2 getNIF() 67
4.16.3.3 getUserName() 68
4.16.3.4 getUserNum() 68
4.16.3.5 operator<() 68
4.16.3.6 showUser() 69
4.16.4 Member Data Documentation 69
4.16.4.1 isAdmin 69
4.16.4.2 name 69
4.16.4.3 NIF 69
4.16.4.4 userNumber 70
4.17 UserNumException Class Reference 70
4.17.1 Detailed Description 71
4.17.2 Constructor & Destructor Documentation 71
4.17.2.1 UserNumException() 71

5 File Documentation **73**
5.1 AirConditionSensor.cpp File Reference 73
5.2 AirConditionSensor.h File Reference 73
5.2.1 Detailed Description 74
5.3 Camera.cpp File Reference 75
5.4 Camera.h File Reference 75
5.4.1 Detailed Description 76
5.5 Dashboard.cpp File Reference 76
5.6 Dashboard.h File Reference 77
5.6.1 Detailed Description 78
5.7 DataBase.cpp File Reference 78
5.8 DataBase.h File Reference 78
5.8.1 Detailed Description 79
5.9 HumiditySensor.cpp File Reference 80
5.10 HumiditySensor.h File Reference 80
5.10.1 Detailed Description 81
5.11 LigthLevelSensor.cpp File Reference 82
5.12 LigthLevelSensor.h File Reference 82

5.12.1 Detailed Description . . . 83
5.13 Login.cpp File Reference . . . 84
5.14 Login.h File Reference . . . 84
5.14.1 Detailed Description . . . 85
5.15 main.cpp File Reference . . . 86
5.15.1 Function Documentation . . . 86
5.15.1.1 main() . . . 87
5.16 Microphone.cpp File Reference . . . 88
5.17 Microphone.h File Reference . . . 88
5.17.1 Detailed Description . . . 89
5.18 NIFException.cpp File Reference . . . 89
5.19 NIFException.h File Reference . . . 90
5.19.1 Detailed Description . . . 90
5.20 RGBCamera.cpp File Reference . . . 91
5.21 RGBCamera.h File Reference . . . 91
5.21.1 Detailed Description . . . 92
5.22 Sensor.cpp File Reference . . . 93
5.23 Sensor.h File Reference . . . 93
5.23.1 Detailed Description . . . 94
5.24 StringException.cpp File Reference . . . 94
5.25 StringException.h File Reference . . . 94
5.25.1 Detailed Description . . . 95
5.26 TemperatureSensor.cpp File Reference . . . 96
5.27 TemperatureSensor.h File Reference . . . 96
5.27.1 Detailed Description . . . 97
5.28 ThermalCamera.cpp File Reference . . . 98
5.29 ThermalCamera.h File Reference . . . 98
5.29.1 Detailed Description . . . 99
5.30 TypeError.cpp File Reference . . . 100
5.31 TypeError.h File Reference . . . 100
5.31.1 Detailed Description . . . 101
5.32 User.cpp File Reference . . . 101
5.33 User.h File Reference . . . 102
5.33.1 Detailed Description . . . 102
5.34 UserNumException.cpp File Reference . . . 103
5.35 UserNumException.h File Reference . . . 103
5.35.1 Detailed Description . . . 104

Index **105**

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

- Camera 12
  - RGBCamera 47
  - ThermalCamera 60
- Dashboard 15
- DataBase 20
- Login 39
- Microphone 42
- std::runtime_error
  - NIFException 46
  - StringException 54
  - TypeError 64
  - UserNumException 70
- Sensor 51
  - AirConditionSensor 7
  - HumiditySensor 29
  - LigthLevelSensor 34
  - TemperatureSensor 55
- User 65

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

AirConditionSensor 7
Camera 12
Dashboard 15
DataBase 20
HumiditySensor 29
LigthLevelSensor 34
Login 39
Microphone 42
NIFException 46
RGBCamera 47
Sensor 51
StringException 54
TemperatureSensor 55
ThermalCamera 60
TypeError 64
User 65
UserNumException 70

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

AirConditionSensor.cpp 73
AirConditionSensor.h
Defines the AirConditionSensor class, wich is is inherited from the sensor class, with its attributes, methods, and constructor 73
Camera.cpp 75
Camera.h
Defines the camera class with its attributes, methods, and constructor 75
Dashboard.cpp 76
Dashboard.h
Defines the dashboard class with its attributes, methods, and constructor 77
DataBase.cpp 78
DataBase.h
Defines the database class with its attributes, methods, and constructor 78
HumiditySensor.cpp 80
HumiditySensor.h
Defines the HumiditySensor class, wich is is inherited from the sensor class, with its attributes, methods, and constructor 80
LigthLevelSensor.cpp 82
LigthLevelSensor.h
Defines the LigthLevelSensor class, wich is is inherited from the sensor class, with its attributes, methods, and constructor 82
Login.cpp 84
Login.h
Defines the login class with its attributes, methods, and constructor 84
main.cpp 86
Microphone.cpp 88
Microphone.h
Defines the microphone class with its attributes, methods, and constructor 88
NIFException.cpp 89
NIFException.h
Defines the NIF exception class with its constructor 90
RGBCamera.cpp 91
RGBCamera.h
Defines the RGBCamera class, wich is is inherited from the camera class, with its attributes, methods, and constructor 91

Sensor.cpp 93
Sensor.h
Defines the sensor class with its attributes, methods, and constructor 93
StringException.cpp 94
StringException.h
Defines the string exception class with its constructor 94
TemperatureSensor.cpp 96
TemperatureSensor.h
Defines the TemperatureSensor class, wich is is inherited from the sensor class, with its attributes, methods, and constructor 96
ThermalCamera.cpp 98
ThermalCamera.h
Defines the ThermalCamera class, wich is is inherited from the camera class, with its attributes, methods, and constructor 98
TypeError.cpp 100
TypeError.h
Defines the user type exception class with its constructor 100
User.cpp 101
User.h
Defines the user class with its attributes, methods, and constructor 102
UserNumException.cpp 103
UserNumException.h
Defines the user number exception class with its constructor 103

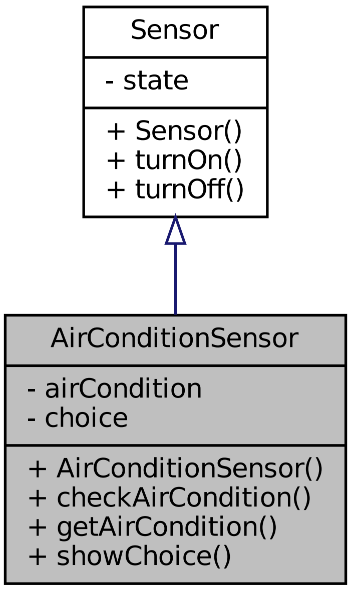
Chapter 4

Class Documentation

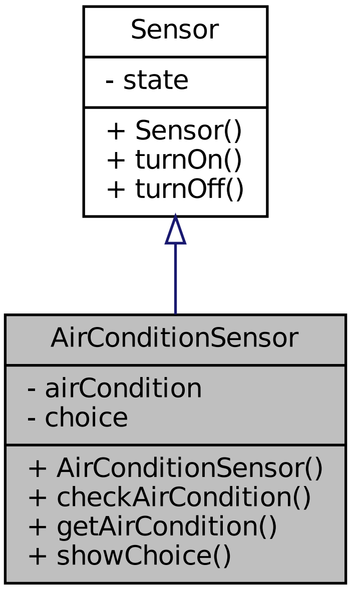
4.1 AirConditionSensor Class Reference

```
#include <AirConditionSensor.h>
```

Inheritance diagram for AirConditionSensor:



Collaboration diagram for AirConditionSensor:



Public Member Functions

- AirConditionSensor ()
- void checkAirCondition ()
- void getAirCondition ()
- void showChoice ()

Private Attributes

- float airCondition
- int choice

4.1.1 Detailed Description

Definition at line 17 of file AirConditionSensor.h.

4.1.2 Constructor & Destructor Documentation

4.1.2.1 AirConditionSensor()

```
AirConditionSensor::AirConditionSensor ( )
```

Definition at line 9 of file AirConditionSensor.cpp.

```
:Sensor() {} //we define the constructor indicating that it is an inherit class of sensor
```

4.1.3 Member Function Documentation

4.1.3.1 checkAirCondition()

```
void AirConditionSensor::checkAirCondition ( )
```

Creates a random number to simulate de current air condition

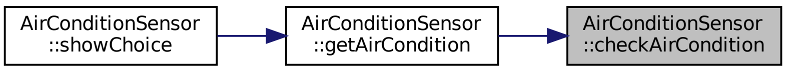
Definition at line 11 of file AirConditionSensor.cpp.

```
                                             {
  //create a random number
  srand(time(NULL)+3);
  airCondition = rand()%301;
}
```

References airCondition.

Referenced by getAirCondition().

Here is the caller graph for this function:


AirConditionSensor ::showChoice
AirConditionSensor ::getAirCondition
AirConditionSensor ::checkAirCondition

4.1.3.2 getAirCondition()

```
void AirConditionSensor::getAirCondition ( )
```

Displays the current ligth level created with checkLigthLevel()

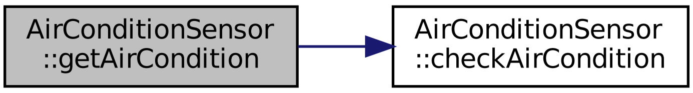
Definition at line 17 of file AirConditionSensor.cpp.

```
                                           {
  //displays the current ligth level created with checkLigthLevel()
  checkAirCondition();
  cout « "\nCurrent air condition level: " « airCondition « " ppm" « endl;
}
```

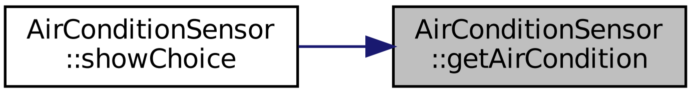
References airCondition, and checkAirCondition().

Referenced by showChoice().

Here is the call graph for this function:


AirConditionSensor ::getAirCondition
AirConditionSensor ::checkAirCondition

Here is the caller graph for this function:


AirConditionSensor ::showChoice
AirConditionSensor ::getAirCondition

4.1.3.3 showChoice()

```
void AirConditionSensor::showChoice ( )
```

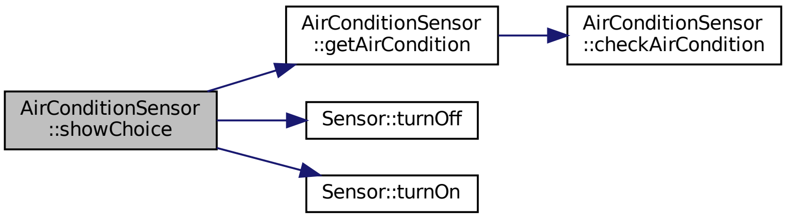
Displays the air condition sensor options menu. Depending on the option that is entered, it calls its respective function.

Definition at line 23 of file AirConditionSensor.cpp.

```
                                      {
//displays the air condition sensor options menu
while (true) {
  cout « "\n                 ___   _                                _  _  _    _               " « endl
  « "                /_ \\ (_)                              | |(_)| |  (_)             " « endl
  « "               / /_\\ \\ _  _ __    ___  ___   _ __    __| | _ | |_  _   ___   _ __  " « endl
  « "               |  _  || || '__|  / __|/ _ \\ | '_ \\  / _` || || __|| | / _ \\ | '_ \\ " « endl
  « "               | | | || || |     | (__| (_) || | | || (_| || || |_ | || (_) || | | |" « endl
  « "               \\_| |_/|_||_|      \\___|\\___/ |_| |_| \\__,_||_| \\__||_| \\___/ |_| |_|" « endl
  « "                                                                                   " « endl
  « "-----------------------------------------------------------------------------------------------" «
  endl
  « "°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°\n"
  « endl;
  cout « "                                    1. TURN ON" « endl
      « "                                    2. TURN OFF" « endl
      « "                                    3. SHOW CURRENT AIR CONDITION" « endl
      « "                                    4. BACK TO MENU" « endl;
  cout « "Enter your choice number: ";
  cin » choice;

  //Depending on the option that is entered, it calls its respective function
  switch (choice)
  {
    case 1:
      Sensor::turnOn();
      system("sleep 3");
      system("clear");
      break;

    case 2:
      Sensor::turnOff();
      system("sleep 3");
      system("clear");
      break;

    case 3:
      getAirCondition();
      system("sleep 3");
      system("clear");
      break;

    case 4:
      //go back to the main menu
      return;
      break;
    }
  }
}
```

References choice, getAirCondition(), Sensor::turnOff(), and Sensor::turnOn().

Here is the call graph for this function:


AirConditionSensor ::showChoice
AirConditionSensor ::getAirCondition
AirConditionSensor ::checkAirCondition
Sensor::turnOff
Sensor::turnOn

4.1.4 Member Data Documentation

4.1.4.1 airCondition

```
float AirConditionSensor::airCondition  [private]
```

Definition at line 41 of file AirConditionSensor.h.

Referenced by checkAirCondition(), and getAirCondition().

4.1.4.2 choice

```
int AirConditionSensor::choice  [private]
```

Definition at line 42 of file AirConditionSensor.h.

Referenced by showChoice().

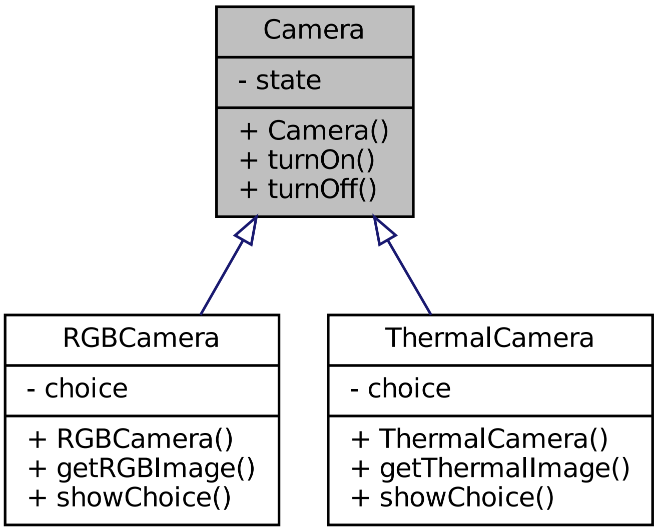
The documentation for this class was generated from the following files:

- AirConditionSensor.h
- AirConditionSensor.cpp

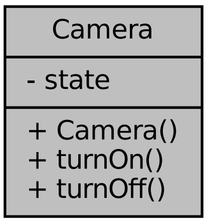
4.2 Camera Class Reference

```
#include <Camera.h>
```

Inheritance diagram for Camera:



Collaboration diagram for Camera:



Public Member Functions

- Camera ()
- void turnOn ()
- void turnOff ()

Private Attributes

- bool state = true

4.2.1 Detailed Description

Definition at line 15 of file Camera.h.

4.2.2 Constructor & Destructor Documentation

4.2.2.1 Camera()

```
Camera::Camera ( )
```

Definition at line 5 of file Camera.cpp.

```
5 {} //we define the constructor
```

4.2.3 Member Function Documentation

4.2.3.1 turnOff()

```
void Camera::turnOff ( )
```

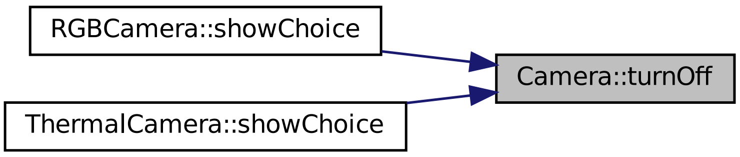
Depending on the state of the camera, we turn it off and indicate its state

Definition at line 17 of file Camera.cpp.

```
                      {
  //Depending on the state of the camera, we turn it off and indicate its state
  if (state == true){
    cout « "\nCamera status: Off" « endl;
    state = false;
  }else{
    cout « "\nThe camera is already off" « endl;
  }
}
```

Referenced by RGBCamera::showChoice(), and ThermalCamera::showChoice().

Here is the caller graph for this function:



4.2.3.2 turnOn()

```
void Camera::turnOn ( )
```

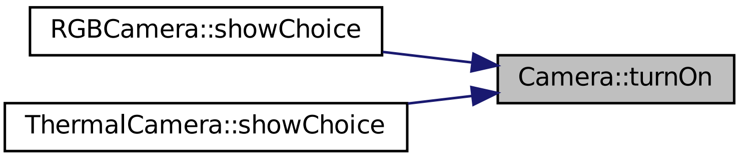
Depending on the state of the camera, we turn it on and indicate its state

Definition at line 7 of file Camera.cpp.

```
                     {
  //Depending on the state of the camera, we turn it on and indicate its state
  if (state == false){
    cout « "\nCamera status: On" « endl;
    state = true;
  }else{
    cout « "\nThe camera is already on" « endl;
  }
}
```

Referenced by RGBCamera::showChoice(), and ThermalCamera::showChoice().

Here is the caller graph for this function:



4.2.4 Member Data Documentation

4.2.4.1 state

```
bool Camera::state = true  [private]
```

Definition at line 32 of file Camera.h.

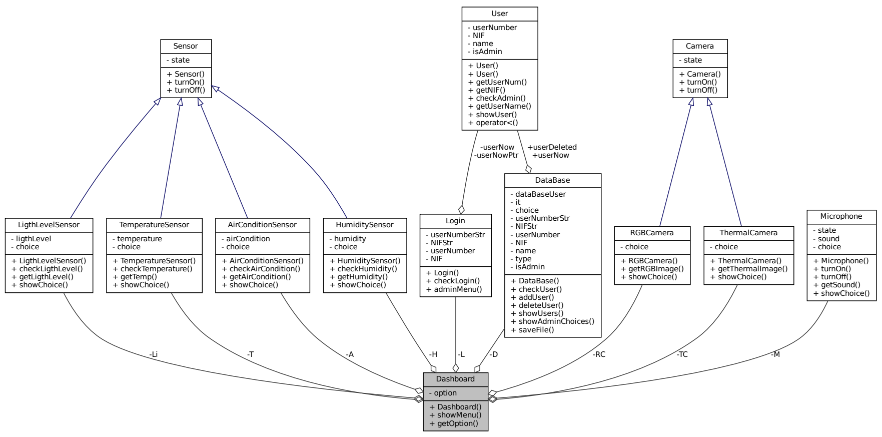
The documentation for this class was generated from the following files:

- Camera.h
- Camera.cpp

4.3 Dashboard Class Reference

```
#include <Dashboard.h>
```

Collaboration diagram for Dashboard:



Public Member Functions

- Dashboard ()
- void showMenu ()
- bool getOption ()

Private Attributes

- int option
- Login L
- DataBase D
- TemperatureSensor T
- HumiditySensor H
- LigthLevelSensor Li
- AirConditionSensor A
- RGBCamera RC
- ThermalCamera TC
- Microphone M

4.3.1 Detailed Description

Definition at line 16 of file Dashboard.h.

4.3.2 Constructor & Destructor Documentation

4.3.2.1 Dashboard()

```
Dashboard::Dashboard ( )
```

Definition at line 18 of file Dashboard.cpp.

```
                       { //we define the constructor
  this -> option = option;
}
```

4.3.3 Member Function Documentation

4.3.3.1 getOption()

```
bool Dashboard::getOption ( )
```

Depending on the option entered, performs an action

Definition at line 22 of file Dashboard.cpp.

```
                            {
  //depending on the option entered, performs an action
  cout « "Enter the number of the action you want to do: ";
  cin » option;

  switch (option)
  {
    case 1:
      system("clear");
      T.showChoice();
      break;

    case 2:
      system("clear");
      H.showChoice();
      break;

    case 3:
      system("clear");
      Li.showChoice();
      break;

    case 4:
      system("clear");
      A.showChoice();
      break;

    case 5:
      system("clear");
      RC.showChoice();
      break;

    case 6:
      system("clear");
      TC.showChoice();
      break;

    case 7:
      system("clear");
      M.showChoice();
      break;

    case 8:
      return true;
      break;

    case 9:
      D.saveFile();
      exit(EXIT_SUCCESS);
      break;

    case 10:
      system("clear");
      D.showAdminChoices();
      break;
  }
  return false;
}
```

4.3.3.2 showMenu()

void Dashboard::showMenu ( )

Displays the login screen and, depending on whether the user exists, displays the main menu Checks that the user entered is of type Admin Allows the showMenu() function to be called again when you want to log out and re-enter a user in case of not putting it, it does not verify that the user exists

Definition at line 81 of file Dashboard.cpp.

```
                           {
    try{
      //displays the login screen and, depending on whether the user exists, displays the main menu
      if (L.checkLogin(&D)){
        while (true){
          system("clear");
          cout « "                __  __    _    ___  _   _       __  __  _____  _   _  _   _ " « endl
          « "               |  \\/  |   / \\   |_ _|| \\ | |     |  \\/  || ____|| \\ | || | | |" « endl
          « "               | |\\/| |  / _ \\   | | |  \\| |     | |\\/| ||  _|  |  \\| || | | |" « endl
          « "               | |  | | / ___ \\  | | | |\\  |     | |  | || |___ | |\\  || |_| |" « endl
          « "               |_|  |_|/_/   \\_\\|___||_| \\_|     |_|  |_||_____||_| \\_| \\___/ " « endl
          « "                                                                              " « endl

          «
    "-----------------------------------------------------------------------------------------------" «
    endl
          «
    "°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°\n" «
    endl;


```

```
        cout « "                                   1. TEMPERATURE SENSOR" « endl « "
                 2. HUMIDITY SENSOR" « endl
        « "                                   3. LIGTH LEVEL SENSOR" « endl « "
             4. AIR CONDITION SENSOR" « endl
        « "                                   5. RGB CAMERA" « endl « "
     6. THERMAL CAMERA" « endl
         « "                                   7. MICROPHONE" « endl « "
      8. SIGN OFF" « endl
         « "                                   9. EXIT" « endl;

         //checks that the user entered is of type Admin
         if(D.userNow.checkAdmin()){
           cout « "                                        10. MANAGE USERS\n" « endl;
         }

         //allows the showMenu() function to be called again when you want to log out and re-enter a
  user
         //in case of not putting it, it does not verify that the user exists
         if (getOption()){
           return;
         }
       }
     }
   }

   catch(UserNumException &except){
     cout « "Exception: " « except.what() « endl;
   }

   catch(NIFException &except){
     cout « "Exception: " « except.what() « endl;
   }
}
```

Referenced by main().

Here is the caller graph for this function:

main
Dashboard::showMenu

4.3.4 Member Data Documentation

4.3.4.1 A

`AirConditionSensor Dashboard::A [private]`

Definition at line 43 of file Dashboard.h.

4.3.4.2 D

`DataBase Dashboard::D [private]`

Definition at line 39 of file Dashboard.h.

4.3.4.3 H

```
HumiditySensor Dashboard::H  [private]
```

Definition at line 41 of file Dashboard.h.

4.3.4.4 L

```
Login Dashboard::L  [private]
```

Definition at line 38 of file Dashboard.h.

4.3.4.5 Li

```
LigthLevelSensor Dashboard::Li  [private]
```

Definition at line 42 of file Dashboard.h.

4.3.4.6 M

```
Microphone Dashboard::M  [private]
```

Definition at line 46 of file Dashboard.h.

4.3.4.7 option

```
int Dashboard::option  [private]
```

Definition at line 36 of file Dashboard.h.

4.3.4.8 RC

```
RGBCamera Dashboard::RC  [private]
```

Definition at line 44 of file Dashboard.h.

4.3.4.9 T

```
TemperatureSensor Dashboard::T  [private]
```

Definition at line 40 of file Dashboard.h.

4.3.4.10 TC

```
ThermalCamera Dashboard::TC  [private]
```

Definition at line 45 of file Dashboard.h.

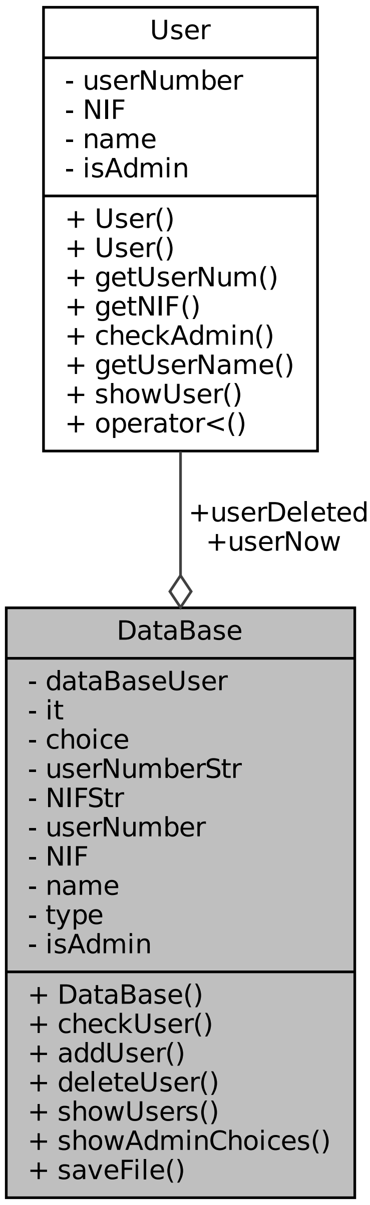
The documentation for this class was generated from the following files:

- Dashboard.h
- Dashboard.cpp

4.4 DataBase Class Reference

```
#include <DataBase.h>
```

Collaboration diagram for DataBase:



Public Member Functions

- DataBase ()
- bool checkUser (int, int)
- void addUser (string, string, string, bool)
- void deleteUser (int)
- void showUsers ()
- void showAdminChoices ()
- void saveFile ()

Public Attributes

- User userNow
- User userDeleted

Private Attributes

- set< User > dataBaseUser
- set< User >::iterator it
- int choice
- std::string userNumberStr
- std::string NIFStr
- int userNumber
- int NIF
- string name
- int type
- bool isAdmin

4.4.1 Detailed Description

Definition at line 18 of file DataBase.h.

4.4.2 Constructor & Destructor Documentation

4.4.2.1 DataBase()

```
DataBase::DataBase ( )
```

Definition at line 17 of file DataBase.cpp.

```
                      { //we define the constructor with the initial users that we are gonna have
  /*this->dataBaseUser.insert(User(1,13172409,"Ana",true));
  this->dataBaseUser.insert(User(2,13172408,"Paula",false));
  this->dataBaseUser.insert(User(3,12345678,"Carlos",false));*/

  // Reads the users found in the users.dat file and dumps them into the database set
  ifstream inUsersFile ("users.dat", ios::in | ios::binary);

  if (!inUsersFile) { // fstream could not open file
    cerr « "File could not be opened." « endl;
    exit (1);
  }

  User user;
  inUsersFile.read (reinterpret_cast <char *>(&user), sizeof (User));
  while (inUsersFile && !inUsersFile.eof()) {
    this->dataBaseUser.insert(user);
    inUsersFile.read (reinterpret_cast <char *>(&user), sizeof (User));
  }
}
```

4.4.3 Member Function Documentation

4.4.3.1 addUser()

```
void DataBase::addUser (
              string userNumberStr,
              string NIFStr,
              string name,
              bool isAdmin )
```

Checks that both the user number and password are the correct size Adds a new user that is entered by an admin in the terminal

Definition at line 55 of file DataBase.cpp.

```
                                                                             {
    //checks that both the user number and password are the correct size
    try{
      for(int n = 0; n < userNumberStr.length(); n++){
        if(int(userNumberStr[n]) < 47 || int(userNumberStr[n] > 57)){
          throw StringException();
        }
        userNumber = stoi(userNumberStr);
      }

      if (userNumber < 1 || userNumber > 99999){
        throw UserNumException();
      }

      for(int n = 0; n < NIFStr.length(); n++){
        if(int(NIFStr[n]) < 47 || int(NIFStr[n] > 57)){
          throw StringException();
        }
        NIF = stoi(NIFStr);
      }

      if (NIF < 9999999 || NIF > 99999999){
        throw NIFException();
      }

      if(isAdmin != 1 && isAdmin != 0){
        throw TypeError();
      }

      //adds a new user that is entered by an admin in the terminal
      this->dataBaseUser.insert(User(userNumber, NIF, name, isAdmin));
    }

    catch(UserNumException &except){
      cout « "Exception: " « except.what() « endl;
    }

    catch(NIFException &except){
      cout « "Exception: " « except.what() « endl;
    }

    catch(TypeError &except){
      cout « "Exception: " « except.what() « endl;
    }

     catch(StringException &except){
       cout « "Exception: " « except.what() « endl;
     }
 }
```

4.4.3.2 checkUser()

```
bool DataBase::checkUser (
              int userNumber,
              int NIF )
```

Checks if the user exixts

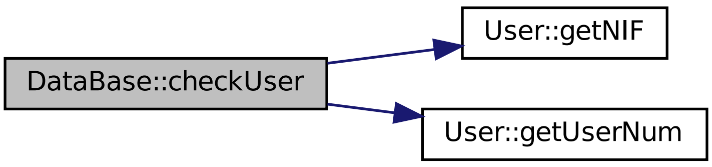
Definition at line 38 of file DataBase.cpp.

```
                                            {
    bool authentication = false;
    //checks if the user exixts
    do{
      for(it=dataBaseUser.begin(); it!=dataBaseUser.end(); it++){
        User user = *it;
        if(userNumber==user.getUserNum() && NIF==user.getNIF()){
            userNow = user;
            return true;
            authentication = true;
        }
      }
    }while(authentication == false);

    return false;
}
```

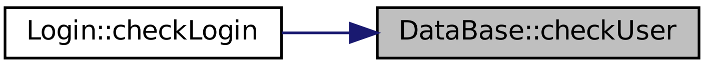
References User::getNIF(), and User::getUserNum().

Referenced by Login::checkLogin().

Here is the call graph for this function:



Here is the caller graph for this function:



4.4.3.3 deleteUser()

```
void DataBase::deleteUser (
            int userNumber )
```

Deletes the user that is entered by an admin in the terminal

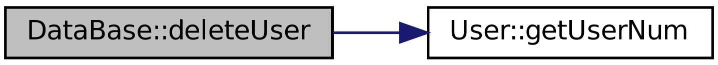
Definition at line 105 of file DataBase.cpp.

```
                                   {
    //deletes the user that is entered by an admin in the terminal
```

```
  for(it=dataBaseUser.begin(); it!=dataBaseUser.end(); it++){
    User user = *it;
    if(userNumber==user.getUserNum()){
      userDeleted = user;
    }
  }
  this->dataBaseUser.erase(userDeleted);
  cout « "El usuario ha sido eliminado" « endl;
}
```

References User::getUserNum().

Here is the call graph for this function:



4.4.3.4 saveFile()

```
void DataBase::saveFile ( )
```

Saves the users found in the set to the file users.dat

Definition at line 181 of file DataBase.cpp.

```
                     {
  fstream outUsersFile ("users.dat", ios::in | ios::out | ios::trunc | ios::binary); // ios::in will
   require an existing file
  // Uses: "users.dat", generated in previous example
  if (!outUsersFile) { // fstream could not open file
    cerr « "File could not be opened." « endl;
    exit (1);
  }

  int position = 0;

  for(it=dataBaseUser.begin(); it!=dataBaseUser.end(); it++){
    User user = *it;
    outUsersFile.seekp (position * sizeof (User));
    outUsersFile.write (reinterpret_cast <const char *> (&user), sizeof (User));
    position ++;
  }
}
```

4.4.3.5 showAdminChoices()

```
void DataBase::showAdminChoices ( )
```

Displays the manage users options menu, that only an Admin can use

Definition at line 130 of file DataBase.cpp.

```
                             {
  //displays the manage users options menu, that only an Admin can use
```

```
132   while (true) {
133     cout « "\n                             MANAGE USERS" « endl
134       « "----------------------------------------------------------------------" « endl;
135     cout « "                    1. ADD USER" « endl
136         « "                    2. DELETE USER" « endl
137         « "                    3. SHOW USERS LIST" « endl
138         « "                    4. BACK TO MENU" « endl;
139     cout « "Enter your choice number: ";
140     cin » choice;
141 
142     switch (choice)
143     {
144     case 1:
145       cout « "Enter a new user" « endl;
146       cout « "User number: ";
147       cin » userNumberStr;
148       cout « "User password: ";
149       cin » NIFStr;
150       cout « "User name: ";
151       cin » name;
152       cout « "User type (1=Admin, 0=User): ";
153       cin » type;
154 
155       if (type == 1){
156         isAdmin = true;
157       }else{  isAdmin = false; }
158 
159       addUser(userNumberStr, NIFStr, name, isAdmin);
160       break;
161 
162     case 2:
163       cout « "Enter the user number that you want to delete: ";
164       cin » userNumber;
165       deleteUser(userNumber);
166       break;
167 
168     case 3:
169       cout « "USERS LIST: " « endl;
170       cout « "Nº\tNAME\t\tTYPE" « endl;
171       showUsers();
172       break;
173 
174     case 4:
175       return;
176       break;
177     }
178   }
179 }
```

4.4.3.6 showUsers()

`void DataBase::showUsers ( )`

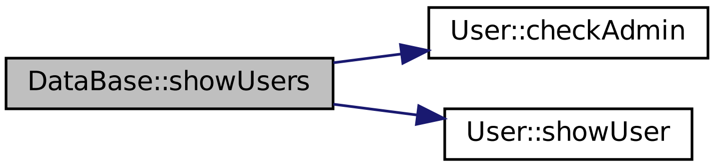
Shows the users that are saved in the database

Definition at line 117 of file DataBase.cpp.

```
117                         {
118   //shows the users that are saved in the database
119   for(it=dataBaseUser.begin(); it!=dataBaseUser.end(); it++){
120     User user = *it;
121     user.showUser();
122     if(user.checkAdmin()){
123       cout « "\t\tAdmin" « endl;
124     }else{
125       cout « "\t\tEmployee" « endl;
126     }
127   }
128 }
```

References User::checkAdmin(), and User::showUser().

Here is the call graph for this function:



4.4.4 Member Data Documentation

4.4.4.1 choice

```
int DataBase::choice  [private]
```

Definition at line 62 of file DataBase.h.

4.4.4.2 dataBaseUser

```
set<User> DataBase::dataBaseUser  [private]
```

Definition at line 60 of file DataBase.h.

4.4.4.3 isAdmin

```
bool DataBase::isAdmin  [private]
```

Definition at line 69 of file DataBase.h.

4.4.4.4 it

```
set<User>::iterator DataBase::it  [private]
```

Definition at line 61 of file DataBase.h.

4.4.4.5 name

```
string DataBase::name  [private]
```

Definition at line 67 of file DataBase.h.

4.4.4.6 NIF

```
int DataBase::NIF  [private]
```

Definition at line 66 of file DataBase.h.

4.4.4.7 NIFStr

```
std::string DataBase::NIFStr  [private]
```

Definition at line 64 of file DataBase.h.

4.4.4.8 type

```
int DataBase::type  [private]
```

Definition at line 68 of file DataBase.h.

4.4.4.9 userDeleted

```
User DataBase::userDeleted
```

Definition at line 56 of file DataBase.h.

4.4.4.10 userNow

```
User DataBase::userNow
```

Definition at line 55 of file DataBase.h.

4.4.4.11 userNumber

```
int DataBase::userNumber  [private]
```

Definition at line 65 of file DataBase.h.

4.4.4.12 userNumberStr

```
std::string DataBase::userNumberStr  [private]
```

Definition at line 63 of file DataBase.h.

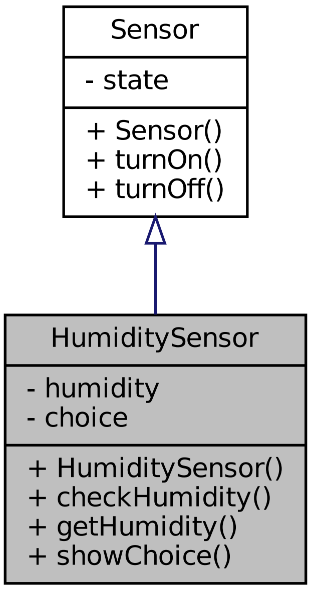
The documentation for this class was generated from the following files:

- DataBase.h
- DataBase.cpp

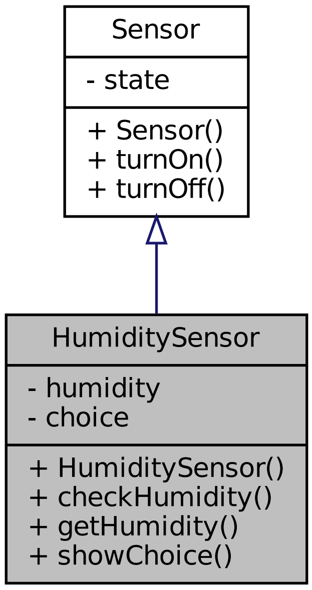
4.5 HumiditySensor Class Reference

```
#include <HumiditySensor.h>
```

Inheritance diagram for HumiditySensor:



Collaboration diagram for HumiditySensor:



Public Member Functions

- HumiditySensor ()
- void checkHumidity ()
- void getHumidity ()
- void showChoice ()

Private Attributes

- float humidity
- int choice

4.5.1 Detailed Description

Definition at line 17 of file HumiditySensor.h.

4.5.2 Constructor & Destructor Documentation

4.5.2.1 HumiditySensor()

```
HumiditySensor::HumiditySensor ( )
```

Definition at line 9 of file HumiditySensor.cpp.

```
:Sensor() {} //we define the constructor indicating that it is an inherit class of sensor
```

4.5.3 Member Function Documentation

4.5.3.1 checkHumidity()

```
void HumiditySensor::checkHumidity ( )
```

Creates a random number to simulate de current humidity

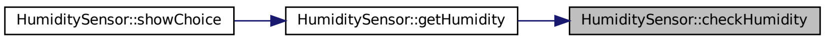
Definition at line 11 of file HumiditySensor.cpp.

```
                                  {
  //create a random number
  srand(time(NULL)+1);
  humidity = rand()%101;
}
```

References humidity.

Referenced by getHumidity().

Here is the caller graph for this function:


HumiditySensor::showChoice
HumiditySensor::getHumidity
HumiditySensor::checkHumidity

4.5.3.2 getHumidity()

```
void HumiditySensor::getHumidity ( )
```

Displays the current humidity created with checkHumidity()

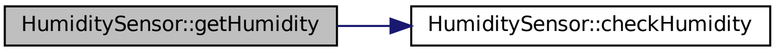
Definition at line 17 of file HumiditySensor.cpp.

```
                                 {
  //displays the current humidity created with checkHumidity()
  checkHumidity();
  cout « "\nCurrent humidity: " « humidity « "%" « endl;
}
```

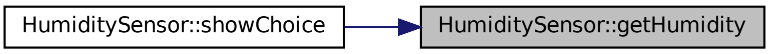
References checkHumidity(), and humidity.

Referenced by showChoice().

Here is the call graph for this function:



Here is the caller graph for this function:



4.5.3.3 showChoice()

```
void HumiditySensor::showChoice ( )
```

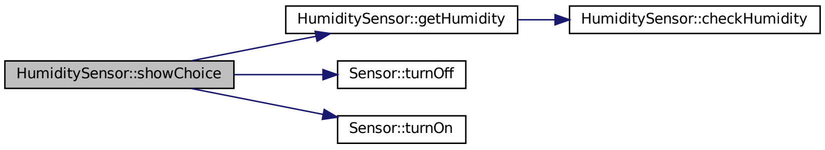
Displays the humidity sensor options menu Depending on the option that is entered, it calls its respective function

Definition at line 23 of file HumiditySensor.cpp.

```
                               {
  //displays the humidity sensor options menu
  while (true) {
    cout « "\n               _    _                    _      _  _  _            " « endl
    « "               | | | |                  (_)     | |(_)| |          " « endl
    « "               | |_| | _   _  _ __ ___   _   __| | _ | |_  _   _ " « endl
    « "               |  _  || | | || '_ ` _ \\ | | / _` || || __|| | | |" « endl
    « "               | | | || |_| || | | | | || || (_| || || |_ | |_| |" « endl
    « "               \\_| |_/ \\__,_||_| |_| |_||_| \\__,_||_| \\__| \\__, |" « endl
    « "                                                           __/ |" « endl
    « "                                                          |___/ " « endl
    « "                                                                " «endl
    « "------------------------------------------------------------------------------------------------" «
    endl
    « "°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°\n"
    « endl;
    cout « "                                   1. TURN ON" « endl
        « "                                  2. TURN OFF" « endl
        « "                                  3. SHOW CURRENT HUMIDITY" « endl
        « "                                  4. BACK TO MENU" « endl;
    cout « "Enter your choice number: ";
    cin » choice;

    //Depending on the option that is entered, it calls its respective function
    switch (choice)
    {
    case 1:
      Sensor::turnOn();
      system("sleep 3");
      system("clear");
      break;

    case 2:
      Sensor::turnOff();
      system("sleep 3");
      system("clear");
      break;

    case 3:
      getHumidity();
      system("sleep 3");
      system("clear");
      break;

    case 4:
      //go back to the main menu
      return;
      break;
    }
  }
}
```

References choice, getHumidity(), Sensor::turnOff(), and Sensor::turnOn().

Here is the call graph for this function:


HumiditySensor::showChoice
HumiditySensor::getHumidity
HumiditySensor::checkHumidity
Sensor::turnOff
Sensor::turnOn

4.5.4 Member Data Documentation

4.5.4.1 choice

`int HumiditySensor::choice [private]`

Definition at line 41 of file HumiditySensor.h.

Referenced by showChoice().

4.5.4.2 humidity

`float HumiditySensor::humidity [private]`

Definition at line 40 of file HumiditySensor.h.

Referenced by checkHumidity(), and getHumidity().

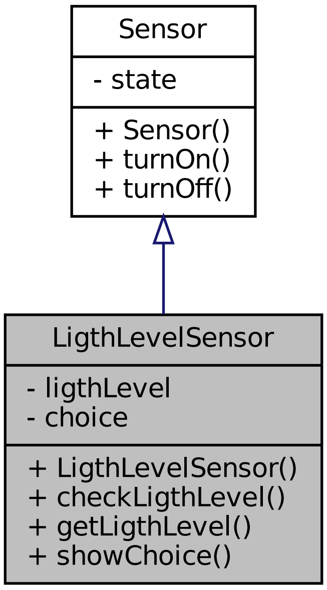
The documentation for this class was generated from the following files:

- HumiditySensor.h
- HumiditySensor.cpp

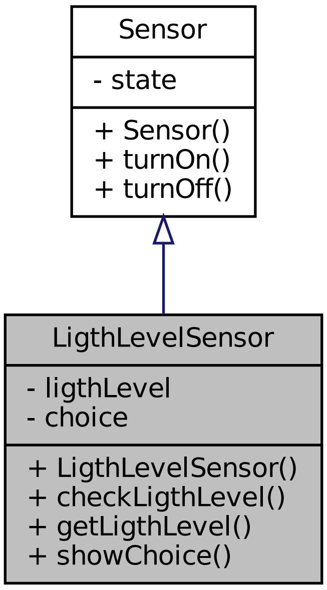
4.6 LigthLevelSensor Class Reference

`#include <LigthLevelSensor.h>`

Inheritance diagram for LigthLevelSensor:



Collaboration diagram for LigthLevelSensor:



Public Member Functions

- LigthLevelSensor ()
- void checkLigthLevel ()
- void getLigthLevel ()
- void showChoice ()

Private Attributes

- float ligthLevel
- int choice

4.6.1 Detailed Description

Definition at line 17 of file LigthLevelSensor.h.

4.6.2 Constructor & Destructor Documentation

4.6.2.1 LigthLevelSensor()

```
LigthLevelSensor::LigthLevelSensor ( )
```

Definition at line 9 of file LigthLevelSensor.cpp.

```
:Sensor() {} //we define the constructor indicating that it is an inherit class of sensor
```

4.6.3 Member Function Documentation

4.6.3.1 checkLigthLevel()

```
void LigthLevelSensor::checkLigthLevel ( )
```

Creates a random number to simulate de current ligth level

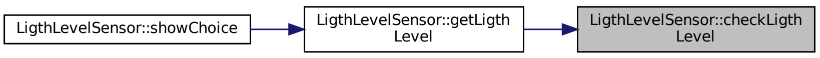
Definition at line 11 of file LigthLevelSensor.cpp.

```
                                     {
  //create a random number
  srand(time(NULL)+2);
  ligthLevel = rand()%101;
}
```

References ligthLevel.

Referenced by getLigthLevel().

Here is the caller graph for this function:


LigthLevelSensor::showChoice
LigthLevelSensor::getLigth Level
LigthLevelSensor::checkLigth Level

4.6.3.2 getLigthLevel()

```
void LigthLevelSensor::getLigthLevel ( )
```

Displays the current ligth level created with checkLigthLevel()

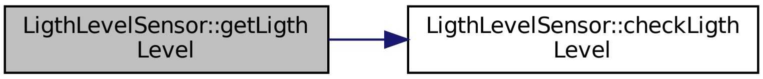
Definition at line 17 of file LigthLevelSensor.cpp.

```
                                    {
  //displays the current ligth level created with checkLigthLevel()
  checkLigthLevel();
  cout « "\nCurrent ligth level: " « ligthLevel « "%" « endl;
}
```

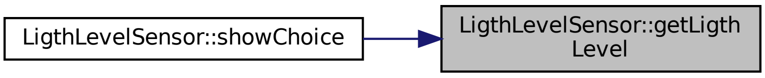
References checkLigthLevel(), and ligthLevel.

Referenced by showChoice().

Here is the call graph for this function:


LigthLevelSensor::getLigth Level
LigthLevelSensor::checkLigth Level

Here is the caller graph for this function:


LigthLevelSensor::showChoice
LigthLevelSensor::getLigth Level

4.6.3.3 showChoice()

```
void LigthLevelSensor::showChoice ( )
```

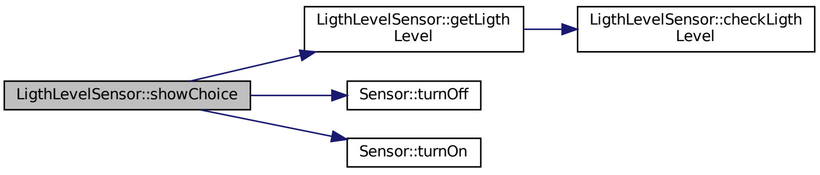
Displays the ligth level sensor options menu Depending on the option that is entered, it calls its respective function

Definition at line 23 of file LigthLevelSensor.cpp.

```
                                    {
    //displays the ligth level sensor options menu
    while (true) {
      cout « "\n                   _      _         _    _       _                    _ " « endl
      « "               | |     (_)       | |  | |     | |                  | |" « endl
      « "               | |      _   __ _ | |_ | |__   | |  ___ __   __ ___ | |" « endl
      « "               | |___| | / _` || __|| '_ \\  | | / _ \\  \\ / // _ \\| |" « endl
      « "               | |_____| || (_| || |_ | | | | ||  __/ \\ V /|  __/| |" « endl
      « "               \\_____/|_| \\__, | \\__||_| |_| |_| \\___|  \\_/  \\___||_|" « endl
      « "                           __/ |                                  " « endl
      « "                          |___/                                   " « endl
      « "                                                                  " « endl
      « "-----------------------------------------------------------------------------------------------" «
      endl
      « "°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°\n"
      « endl;
      cout « "                                 1. TURN ON" « endl
          « "                                2. TURN OFF" « endl
          « "                                3. SHOW CURRENT LIGTH LEVEL" « endl
          « "                                4. BACK TO MENU" « endl;
      cout « "Enter your choice number: ";
      cin » choice;

      //Depending on the option that is entered, it calls its respective function
      switch (choice)
      {
      case 1:
        Sensor::turnOn();
        system("sleep 3");
        system("clear");
        break;

      case 2:
        Sensor::turnOff();
        system("sleep 3");
        system("clear");
        break;

      case 3:
        getLigthLevel();
        system("sleep 3");
        system("clear");
        break;

      case 4:
        //go back to the main menu
        return;
        break;
      }
    }
  }
```

References choice, getLigthLevel(), Sensor::turnOff(), and Sensor::turnOn().

Here is the call graph for this function:


LigthLevelSensor::showChoice
LigthLevelSensor::getLigth Level
LigthLevelSensor::checkLigth Level
Sensor::turnOff
Sensor::turnOn

4.6.4 Member Data Documentation

4.6.4.1 choice

`int LigthLevelSensor::choice [private]`

Definition at line 41 of file LigthLevelSensor.h.

Referenced by showChoice().

4.6.4.2 ligthLevel

`float LigthLevelSensor::ligthLevel [private]`

Definition at line 40 of file LigthLevelSensor.h.

Referenced by checkLigthLevel(), and getLigthLevel().

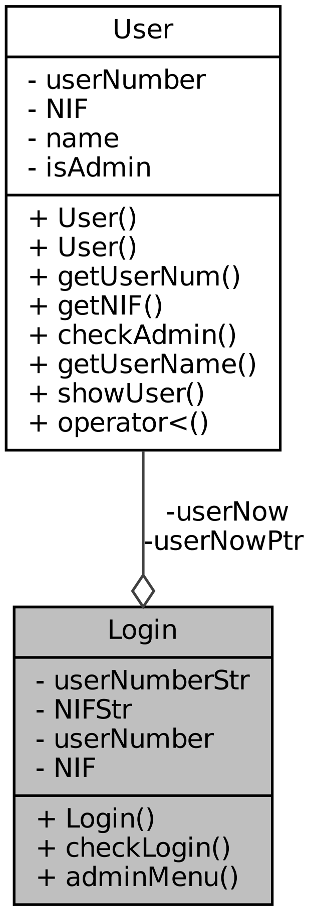
The documentation for this class was generated from the following files:

- LigthLevelSensor.h
- LigthLevelSensor.cpp

4.7 Login Class Reference

```
#include <Login.h>
```

Collaboration diagram for Login:



Public Member Functions

- Login ()
- bool checkLogin (DataBase ∗)
- void adminMenu (DataBase ∗)

Private Attributes

- std::string userNumberStr
- std::string NIFStr
- int userNumber
- int NIF
- User userNow
- User ∗ userNowPtr

4.7.1 Detailed Description

Definition at line 17 of file Login.h.

4.7.2 Constructor & Destructor Documentation

4.7.2.1 Login()

```
Login::Login ( )
```

Definition at line 8 of file Login.cpp.

```
8 {} //we define the constructor
```

4.7.3 Member Function Documentation

4.7.3.1 adminMenu()

```
void Login::adminMenu (
              DataBase * )
```

Show the extra admin option in case the user entered is an admin

4.7.3.2 checkLogin()

```
bool Login::checkLogin (
              DataBase * d )
```

Returns if the login is correct

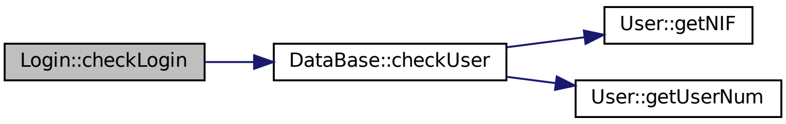
Definition at line 10 of file Login.cpp.

```
10                                    {
11   bool excep = true;
12   while (excep == true){
13     try{
14       cout«"\n                               User: ";
15       cin»userNumberStr;
16       for(int n = 0; n < userNumberStr.length(); n++){
17         if(int(userNumberStr[n]) < 47 || int(userNumberStr[n] > 57)){
18           throw StringException();
19           excep = false;
20         }
21         userNumber = stoi(userNumberStr);
22       }
23 
24       //returns the password entered
25       cout«"                              Password: ";
26       cin»NIFStr;
27       for(int n = 0; n < NIFStr.length(); n++){
28         if(int(NIFStr[n]) < 47 || int(NIFStr[n] > 57)){
29           throw StringException();
30           excep = false;
31         }
32         NIF = stoi(NIFStr);
```

```
      }
    }

    catch(StringException &except){
      cout « "Exception: " « except.what() « endl;
      continue;
    }

    return d->checkUser(userNumber, NIF);
  }
}
```

References DataBase::checkUser().

Here is the call graph for this function:



4.7.4 Member Data Documentation

4.7.4.1 NIF

`int Login::NIF [private]`

Definition at line 38 of file Login.h.

4.7.4.2 NIFStr

`std::string Login::NIFStr [private]`

Definition at line 36 of file Login.h.

4.7.4.3 userNow

`User Login::userNow [private]`

Definition at line 39 of file Login.h.

4.7.4.4 userNowPtr

```
User* Login::userNowPtr  [private]
```

Definition at line 40 of file Login.h.

4.7.4.5 userNumber

```
int Login::userNumber  [private]
```

Definition at line 37 of file Login.h.

4.7.4.6 userNumberStr

```
std::string Login::userNumberStr  [private]
```

Definition at line 35 of file Login.h.

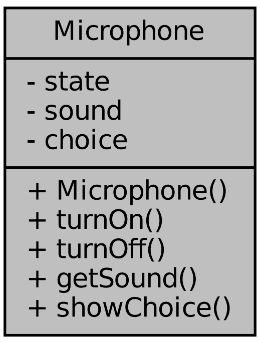
The documentation for this class was generated from the following files:

- Login.h
- Login.cpp

4.8 Microphone Class Reference

```
#include <Microphone.h>
```

Collaboration diagram for Microphone:



Public Member Functions

- Microphone ()
- void turnOn ()
- void turnOff ()
- void getSound ()
- void showChoice ()

Private Attributes

- bool state = true
- string sound = "\nListening command..."
- int choice

4.8.1 Detailed Description

Definition at line 15 of file Microphone.h.

4.8.2 Constructor & Destructor Documentation

4.8.2.1 Microphone()

```
Microphone::Microphone ( )
```

Definition at line 7 of file Microphone.cpp.

```
{} //we define the constructor
```

4.8.3 Member Function Documentation

4.8.3.1 getSound()

```
void Microphone::getSound ( )
```

Prints that a command is being listened for

Definition at line 9 of file Microphone.cpp.

```
                            {
  //prints that a command is being listened for
  cout « sound « endl;
}
```

4.8.3.2 showChoice()

```
void Microphone::showChoice ( )
```

Displays the microphone options menu Depending on the option that is entered, it calls its respective function.

Definition at line 34 of file Microphone.cpp.

```
34                             {
35   //displays the microphone options menu
36   while (true) {
37     cout « "\n              ___  ___ _                         _                          " « endl
38     « "               |  \\/  |(_)                          | |                        " « endl
39     « "               | .  . | _   ___  _ __  ___   _ __  | |__    ___   _ __    ___ " « endl
40     « "               | |\\/| || | / __|| '__|/ _ \\ | '_ \\ | '_ \\  / _ \\ | '_ \\  / _ \\" « endl
41     « "               | |  | || || (__ | |   | (_) || |_) || | | || (_) || | | ||  __/" « endl
42     « "               \\_|  |_/|_| \\___||_|   \\___/ | .__/ |_| |_| \\___/ |_| |_| \\___|" « endl
43     « "                                             | |                                 " « endl
44     « "                                             |_|                                 " « endl
45     « "                                                                                 " « endl
46     « "-----------------------------------------------------------------------------------------------" «
       endl
47     « "°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°\n"
       « endl;
48     cout « "                                 1. TURN ON" « endl
49         « "                                2. TURN OFF" « endl
50         « "                                3. RECORD COMMAND" « endl
51         « "                                4. BACK TO MENU" « endl;
52     cout « "Enter your choice number: ";
53     cin » choice;
54 
55     switch (choice)
56     {
57     case 1:
58       turnOn();
59       system("sleep 3");
60       system("clear");
61       break;
62 
63     case 2:
64       turnOff();
65       system("sleep 3");
66       system("clear");
67       break;
68 
69     case 3:
70       getSound();
71       system("sleep 3");
72       system("clear");
73       break;
74 
75     case 4:
76       return;
77       break;
78     }
79   }
80 }
```

4.8.3.3 turnOff()

```
void Microphone::turnOff ( )
```

Depending on the state of the microphone, we turn it off and indicate its state

Definition at line 24 of file Microphone.cpp.

```
24                            {
25   //Depending on the state of the microphone, we turn it off and indicate its state
26   if (state = true){
27     cout « "\nMicrophone status: Off" « endl;
28     state = false;
29   }else{
30     cout « "\nThe microphone is already off" « endl;
31   }
32 }
```

4.8.3.4 turnOn()

```
void Microphone::turnOn ( )
```

Depending on the state of the microphone, we turn it on and indicate its state

Definition at line 14 of file Microphone.cpp.

```
                          {
  //Depending on the state of the microphone, we turn it on and indicate its state
  if (state == false){
    cout « "\nMicrophone status: On" « endl;
    state = true;
  }else{
    cout « "\nThe microphone is already on" « endl;
  }
}
```

4.8.4 Member Data Documentation

4.8.4.1 choice

```
int Microphone::choice  [private]
```

Definition at line 45 of file Microphone.h.

4.8.4.2 sound

```
string Microphone::sound = "\nListening command..."  [private]
```

Definition at line 44 of file Microphone.h.

4.8.4.3 state

```
bool Microphone::state = true  [private]
```

Definition at line 43 of file Microphone.h.

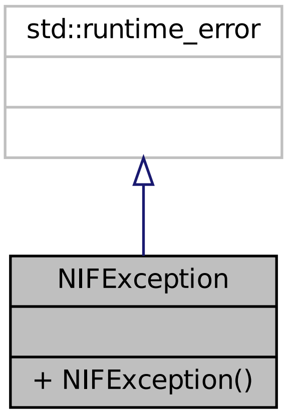
The documentation for this class was generated from the following files:

- Microphone.h
- Microphone.cpp

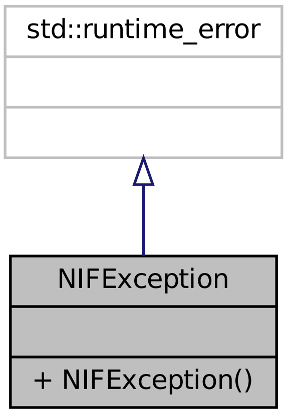
4.9 NIFException Class Reference

```
#include <NIFException.h>
```

Inheritance diagram for NIFException:



Collaboration diagram for NIFException:



Public Member Functions

- NIFException ()

4.9.1 Detailed Description

Definition at line 9 of file NIFException.h.

4.9.2 Constructor & Destructor Documentation

4.9.2.1 NIFException()

```
NIFException::NIFException ( )
```

Definition at line 3 of file NIFException.cpp.

```
:std::runtime_error ("the number of digits for password must be 8."){};
```

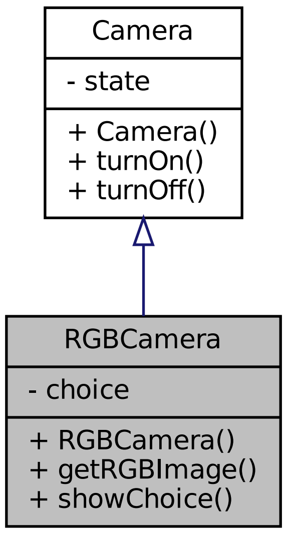
The documentation for this class was generated from the following files:

- NIFException.h
- NIFException.cpp

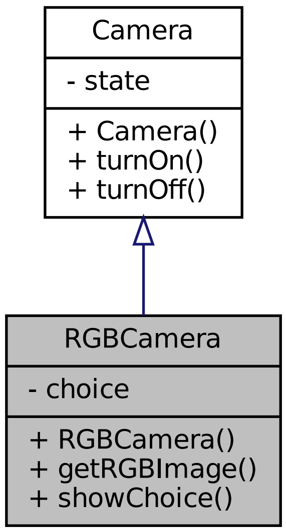
4.10 RGBCamera Class Reference

```
#include <RGBCamera.h>
```

Inheritance diagram for RGBCamera:



Collaboration diagram for RGBCamera:



Public Member Functions

- RGBCamera ()
- void getRGBImage ()
- void showChoice ()

Private Attributes

- int choice

4.10.1 Detailed Description

Definition at line 17 of file RGBCamera.h.

4.10.2 Constructor & Destructor Documentation

4.10.2.1 RGBCamera()

```
RGBCamera::RGBCamera ( )
```

Definition at line 9 of file RGBCamera.cpp.

```
9 :Camera() {} //we define the constructor indicating that it is an inherit class of camera
```

4.10.3 Member Function Documentation

4.10.3.1 getRGBImage()

```
void RGBCamera::getRGBImage ( )
```

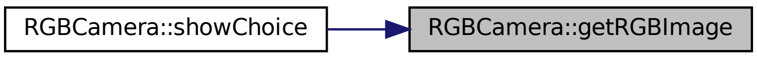
Prints that an image is being displayed

Definition at line 11 of file RGBCamera.cpp.

```
                           {
//prints that an image is being displayed
cout « " ---------------------------------------------" « endl « "-
  LIVE... -" « endl « "-                                             -" « endl « "-
                          -" « endl « "-                                             -" « endl « "-
                                          -" « endl « "-                                             -"
  « endl « "-                                             -" « endl « "-
               -" « endl « "-                 RGB image                   -" « endl « "-
                               -" « endl « "-                                             -" « endl «
  "-                                             -" « endl « "-
      -" « endl « "-                                             -" « endl « "-
                      -" « endl « "-                                             -" « endl « "-
                                      -" « endl « " ---------------------------------------------"«
  endl;;
}
```

Referenced by showChoice().

Here is the caller graph for this function:



4.10.3.2 showChoice()

```
void RGBCamera::showChoice ( )
```

Displays the RGB camera options menu
Depending on the option that is entered, it calls its respective function.

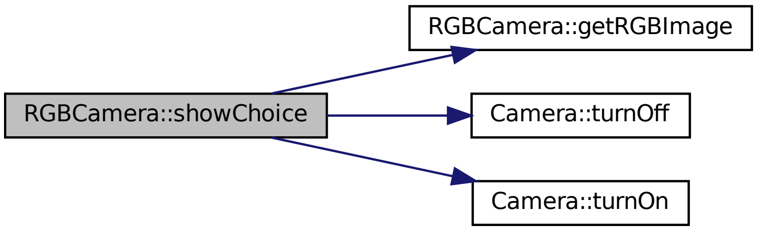
Definition at line 16 of file RGBCamera.cpp.

```
                         {
//displays the RGB camera options menu
while (true) {
  cout « "\n             ______  _____ ______   _____                                                  " « endl
  « "               | ___ \\|  __ \\| ___ \\ /  __ \\                                                " « endl
  « "               | |_/ /| |  \\/| |_/ / | /  \\/  __ _  _ __ ___    ___  _ __  __ _  " « endl
  « "               |    / | | __ | ___ \\ | |     / _` || '_ ` _ \\  / _ \\| '__|/ _` |" « endl
  « "               | |\\ \\ | |_\\ \\| |_/ / | \\__/\\| (_| || | | | | ||  __/| |   | (_| |" « endl
  « "               \\_| \\_| \\____/\\____/   \\____/ \\__,_||_| |_| |_| \\___||_|    \\__,_|" « endl
  « "                                                                            " « endl
```

```
    « "-----------------------------------------------------------------------------------------------" «
    endl
    « "°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°\n"
    « endl;
    cout « "                                    1. TURN ON" « endl
        « "                                    2. TURN OFF" « endl
        « "                                    3. SHOW CURRENT RGB IMAGE" « endl
        « "                                    4. BACK TO MENU" « endl;
    cout « "Enter your choice number: ";
    cin » choice;

    switch (choice)
    {
    case 1:
      Camera::turnOn();
      system("sleep 3");
      system("clear");
      break;

    case 2:
      Camera::turnOff();
      system("sleep 3");
      system("clear");
      break;

    case 3:
      getRGBImage();
      system("sleep 3");
      system("clear");
      break;

    case 4:
      //go back to the main menu
      return;
      break;
    }
  }
}
```

References choice, getRGBImage(), Camera::turnOff(), and Camera::turnOn().

Here is the call graph for this function:



4.10.4 Member Data Documentation

4.10.4.1 choice

`int RGBCamera::choice [private]`

Definition at line 35 of file RGBCamera.h.

Referenced by showChoice().

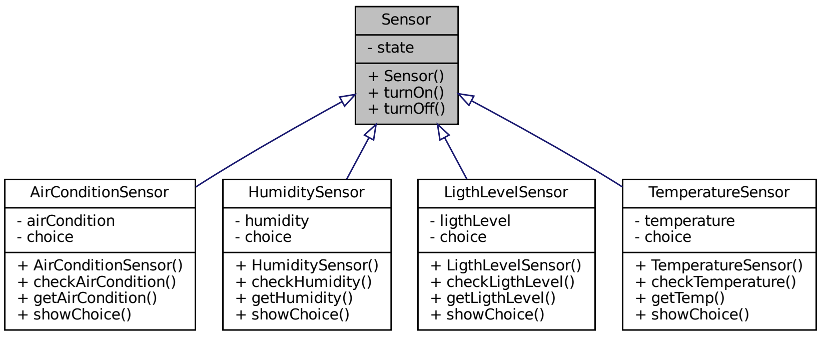
The documentation for this class was generated from the following files:

- RGBCamera.h
- RGBCamera.cpp

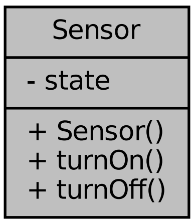
4.11 Sensor Class Reference

```
#include <Sensor.h>
```

Inheritance diagram for Sensor:



Collaboration diagram for Sensor:



Public Member Functions

- Sensor ()
- void turnOn ()
- void turnOff ()

Private Attributes

- bool state = true

4.11.1 Detailed Description

Definition at line 15 of file Sensor.h.

4.11.2 Constructor & Destructor Documentation

4.11.2.1 Sensor()

```
Sensor::Sensor ( )
```

Definition at line 5 of file Sensor.cpp.

```
{} //we define the constructor
```

4.11.3 Member Function Documentation

4.11.3.1 turnOff()

```
void Sensor::turnOff ( )
```

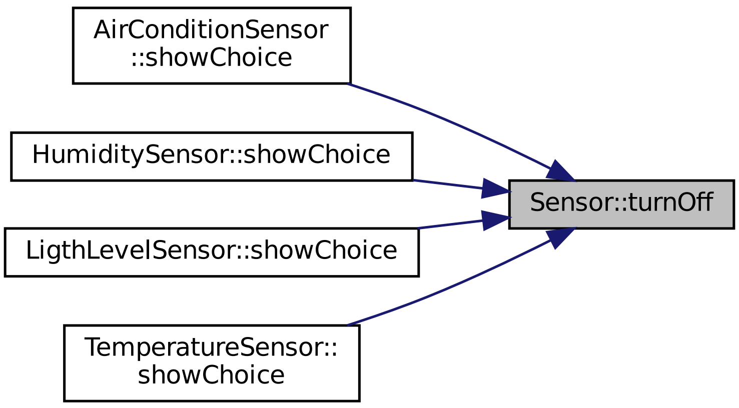
Depending on the state of the sensor, we turn it off and indicate its state

Definition at line 17 of file Sensor.cpp.

```
                      {
    //Depending on the state of the sensor, we turn it off and indicate its state
    if (state == true){
      cout « "\nSensor Status: Off" « endl;
      state = false;
    }else{
      cout « "\nThe sensor is already off" « endl;
    }
}
```

Referenced by AirConditionSensor::showChoice(), HumiditySensor::showChoice(), LigthLevelSensor::show↩ Choice(), and TemperatureSensor::showChoice().

Here is the caller graph for this function:


AirConditionSensor ::showChoice
HumiditySensor::showChoice
LigthLevelSensor::showChoice
TemperatureSensor:: showChoice
Sensor::turnOff

4.11.3.2 turnOn()

```
void Sensor::turnOn ( )
```

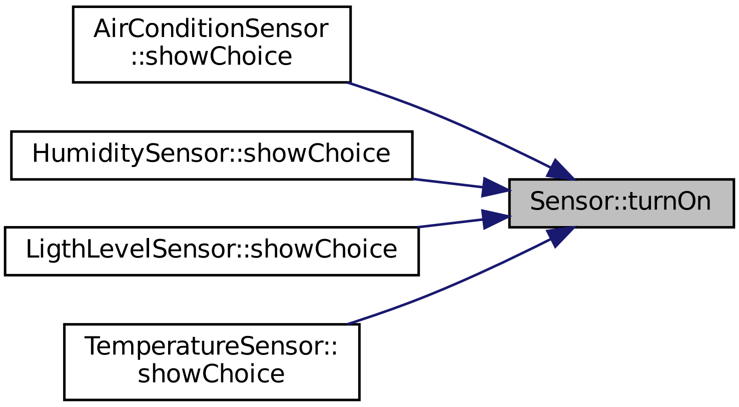
Depending on the state of the sensor, we turn it on and indicate its state

Definition at line 7 of file Sensor.cpp.

```
                      {
  //Depending on the state of the sensor, we turn it on and indicate its state
  if (state == false){
     cout « "\nSensor status: On" « endl;
     state = true;
   }else{
     cout « "\nThe sensor is already on" « endl;
   }
}
```

Referenced by AirConditionSensor::showChoice(), HumiditySensor::showChoice(), LigthLevelSensor::show←
Choice(), and TemperatureSensor::showChoice().

Here is the caller graph for this function:



4.11.4 Member Data Documentation

4.11.4.1 state

```
bool Sensor::state = true  [private]
```

Definition at line 32 of file Sensor.h.

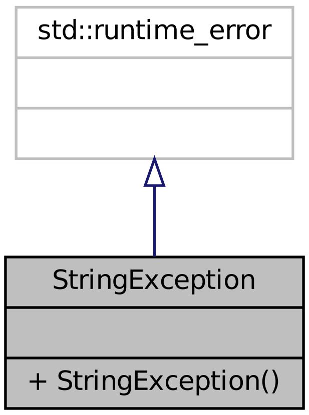
The documentation for this class was generated from the following files:

- Sensor.h
- Sensor.cpp

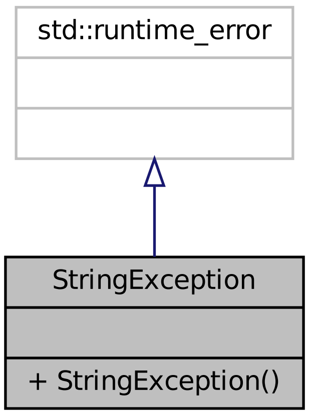
4.12 StringException Class Reference

```
#include <StringException.h>
```

Inheritance diagram for StringException:



Collaboration diagram for StringException:



Public Member Functions

- StringException ()

4.12.1 Detailed Description

Definition at line 9 of file StringException.h.

4.12.2 Constructor & Destructor Documentation

4.12.2.1 StringException()

```
StringException::StringException ( )
```

Definition at line 3 of file StringException.cpp.

```
4   :std::runtime_error ("you've entered a character when only integers are allowed"){};
```

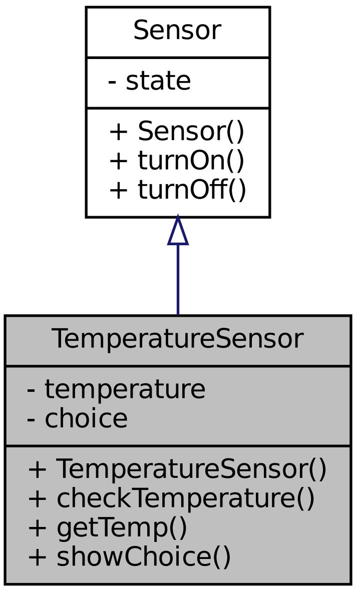
The documentation for this class was generated from the following files:

- StringException.h
- StringException.cpp

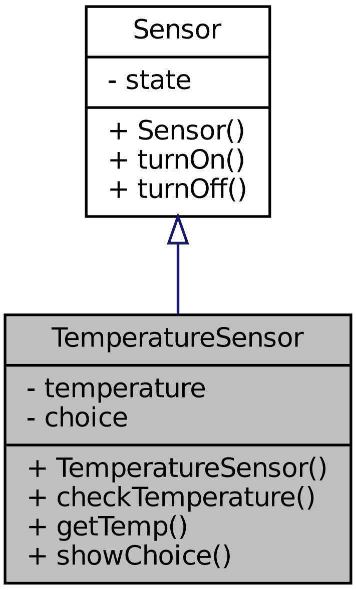
4.13 TemperatureSensor Class Reference

```
#include <TemperatureSensor.h>
```

Inheritance diagram for TemperatureSensor:



Collaboration diagram for TemperatureSensor:



Public Member Functions

- TemperatureSensor ()
- void checkTemperature ()
- void getTemp ()
- void showChoice ()

Private Attributes

- float temperature
- int choice

4.13.1 Detailed Description

Definition at line 17 of file TemperatureSensor.h.

4.13.2 Constructor & Destructor Documentation

4.13.2.1 TemperatureSensor()

```
TemperatureSensor::TemperatureSensor ( )
```

Definition at line 9 of file TemperatureSensor.cpp.

```
:Sensor() {} //we define the constructor indicating that it is an inherit class of sensor
```

4.13.3 Member Function Documentation

4.13.3.1 checkTemperature()

```
void TemperatureSensor::checkTemperature ( )
```

Creates a random number to simulate de current temperature

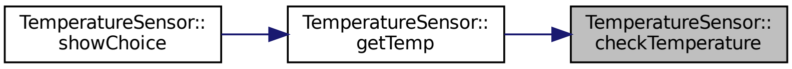
Definition at line 11 of file TemperatureSensor.cpp.

```
                                         {
    //create a random number
    srand(time(NULL));
    temperature = 25+rand()%(40 +1 - 25);
}
```

References temperature.

Referenced by getTemp().

Here is the caller graph for this function:


TemperatureSensor:: showChoice
TemperatureSensor:: getTemp
TemperatureSensor:: checkTemperature

4.13.3.2 getTemp()

```
void TemperatureSensor::getTemp ( )
```

Displays the current temperature created with checkTemperature()

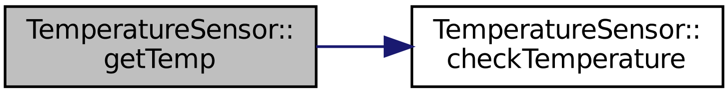
Definition at line 17 of file TemperatureSensor.cpp.

```
                                {
    //displays the current temperature created with checkTemperature()
    checkTemperature();
    cout « "\nCurrent temperature: " « temperature « " °C" « endl;
}
```

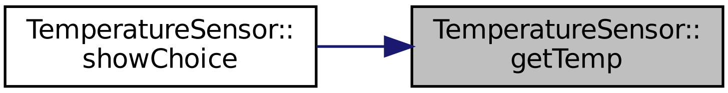
References checkTemperature(), and temperature.

Referenced by showChoice().

Here is the call graph for this function:



Here is the caller graph for this function:



4.13.3.3 showChoice()

```
void TemperatureSensor::showChoice ( )
```

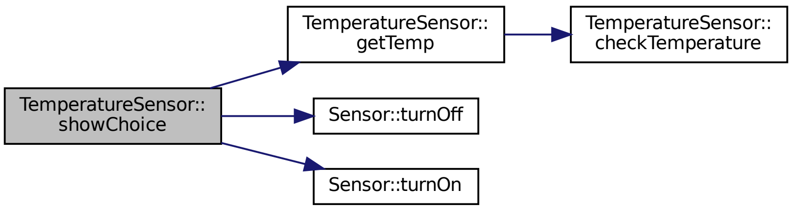
Displays the temperature sensor options menu Depending on the option that is entered, it calls its respective function

Definition at line 23 of file TemperatureSensor.cpp.

```
                                   {
    //displays the temperature sensor options menu
    while (true) {
      cout « "\n                _____                                          _                     " «
      endl
      « "               |_   _|                                          | |                     " « endl
      « "                 | |  ___  _ __ ___   _ __    ___  _ __ __ _ | |_  _   _  _ __  ___ " « endl
      « "                 | | / _ \\| '_ ` _ \\| '_ \\  / _ \\| '__|/ _` || __|| | | || '__|/ _ \\" « endl

      « "                 | ||  __/| | | | | || |_) ||  __/| |   | (_| || |_ | |_| || |   |  __/" « endl
      « "                 \\_/ \\___||_| |_| |_|| .__/  \\___||_|    \\__,_| \\__| \\__,_||_|    \\___|" «
      endl
      « "                                   | |                                          " « endl
      « "                                   |_|                                          " « endl
      « "                                                                               " « endl
      « "------------------------------------------------------------------------------------------------" «
      endl
      « "°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°\n"
      « endl;
      cout « "                                   1. TURN ON" « endl
           « "                                  2. TURN OFF" « endl
           « "                                  3. SHOW CURRENT TEMPERATURE" « endl
           « "                                  4. BACK TO MENU" « endl;
      cout « "\nEnter your choice number: ";
      cin » choice;

      //Depending on the option that is entered, it calls its respective function
      switch (choice)
      {
      case 1:
        Sensor::turnOn();
        system("sleep 3");
        system("clear");
        break;

      case 2:
        Sensor::turnOff();
        system("sleep 3");
        system("clear");
        break;

      case 3:
        getTemp();
        system("sleep 3");
        system("clear");
        break;

      case 4:
        //go back to the main menu
        return;
        break;
      }
    }
  }
```

References choice, getTemp(), Sensor::turnOff(), and Sensor::turnOn().

Here is the call graph for this function:


TemperatureSensor:: showChoice
TemperatureSensor:: getTemp
TemperatureSensor:: checkTemperature
Sensor::turnOff
Sensor::turnOn

4.13.4 Member Data Documentation

4.13.4.1 choice

```
int TemperatureSensor::choice  [private]
```

Definition at line 42 of file TemperatureSensor.h.

Referenced by showChoice().

4.13.4.2 temperature

```
float TemperatureSensor::temperature  [private]
```

Definition at line 41 of file TemperatureSensor.h.

Referenced by checkTemperature(), and getTemp().

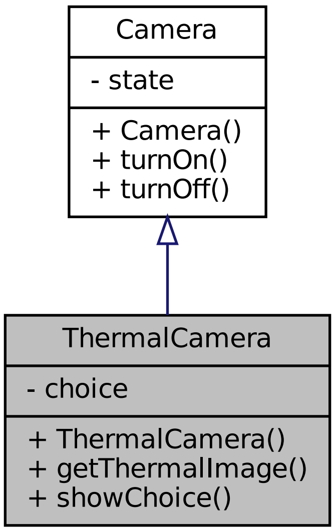
The documentation for this class was generated from the following files:

- TemperatureSensor.h
- TemperatureSensor.cpp

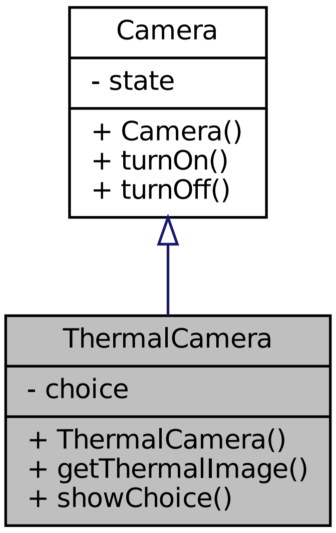
4.14 ThermalCamera Class Reference

```
#include <ThermalCamera.h>
```

Inheritance diagram for ThermalCamera:



Collaboration diagram for ThermalCamera:



Public Member Functions

- ThermalCamera ()
- void getThermalImage ()
- void showChoice ()

Private Attributes

- int choice

4.14.1 Detailed Description

Definition at line 17 of file ThermalCamera.h.

4.14.2 Constructor & Destructor Documentation

4.14.2.1 ThermalCamera()

```
ThermalCamera::ThermalCamera ( )
```

Definition at line 8 of file ThermalCamera.cpp.

```
8 :Camera() {} //we define the constructor indicating that it is an inherit class of camera
```

4.14.3 Member Function Documentation

4.14.3.1 getThermalImage()

```
void ThermalCamera::getThermalImage ( )
```

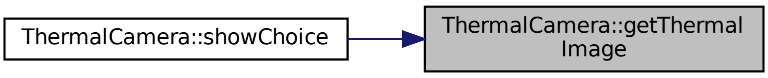
Prints that an image is being displayed

Definition at line 10 of file ThermalCamera.cpp.

```
10                                         {
11   //prints that an image is being displayed
12   cout « " ---------------------------------------------" « endl « "-
       LIVE... -" « endl « "-                                              -" « endl « "-
                         -" « endl « "-                                              -" « endl « "-
                                         -" « endl « "-                                              -"
       « endl « "-                                              -" « endl « "-
                  -" « endl « "-                Thermal image                -" « endl « "-
                                  -" « endl « "-                                              -" « endl «
       "-                                              -" « endl « "-
           -" « endl « "-                                              -" « endl « "-
                           -" « endl « "-                                              -" « endl « "-
                                           -" « endl « " ---------------------------------------------"«
       endl;
13 }
```

Referenced by showChoice().

Here is the caller graph for this function:



4.14.3.2 showChoice()

```
void ThermalCamera::showChoice ( )
```

Displays the thrermal camera options menu Depending on the option that is entered, it calls its respective function

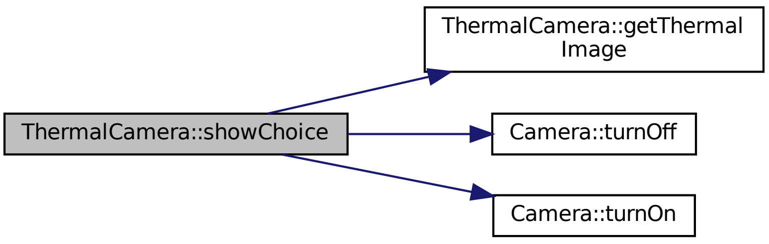
Definition at line 15 of file ThermalCamera.cpp.

```
15                                  {
16   //displays the thrermal camera options menu
17   while (true) {
18     cout « "\n  _____  _                                     _     _____
         " « endl
19     « " |_   _|| |                                   | |   /  __ \\                                " «
       endl
20     « "   | |  | |__    ___  _ __  _ __ ___    __ _ | |   | /  \\/  __ _  _ __ ___    ___  _ __   __ _ " «
       endl
21     « "   | |  | '_ \\  / _ \\| '__|| '_ ` _ \\  / _` || |   | |     / _` || '_ ` _ \\  / _ \\| '__|/ _`
       |" « endl
22     « "   | |  | | | ||  __/| |   | | | | | || (_| || |   | \\__/\\| (_| || | | | | ||  __/| |   | (_| |"
       « endl
```

```
    « "   \\_/  |_| |_| \\___||_|   |_| |_| |_| \\__,_||_|    \\____/ \\__,_||_| |_| |_| \\___||_|
    \\__,_|" « endl
    « "
               " « endl
    « "-----------------------------------------------------------------------------------------------" «
    endl
    « "°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°\n"
    « endl;
    cout « "                                   1. TURN ON" « endl
        « "                                  2. TURN OFF" « endl
        « "                                  3. SHOW CURRENT THERMAL IMAGE" « endl
        « "                                  4. BACK TO MENU" « endl;
    cout « "Enter your choice number: ";
    cin » choice;

    switch (choice)
    {
    case 1:
      Camera::turnOn();
      system("sleep 3");
      system("clear");
      break;

    case 2:
      Camera::turnOff();
      system("sleep 3");
      system("clear");
      break;

    case 3:
      getThermalImage();
      system("sleep 10");
      system("clear");
      break;

    case 4:
      //go back to the main menu
      return;
      break;
    }
  }
}
```

References choice, getThermalImage(), Camera::turnOff(), and Camera::turnOn().

Here is the call graph for this function:



4.14.4 Member Data Documentation

4.14.4.1 choice

`int ThermalCamera::choice [private]`

Definition at line 35 of file ThermalCamera.h.

Referenced by showChoice().

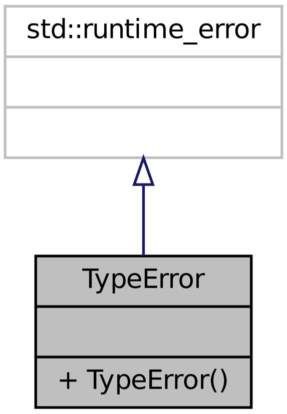
The documentation for this class was generated from the following files:

- ThermalCamera.h
- ThermalCamera.cpp

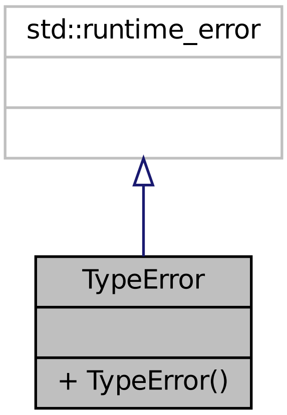
4.15 TypeError Class Reference

```
#include <TypeError.h>
```

Inheritance diagram for TypeError:



Collaboration diagram for TypeError:



Public Member Functions

- TypeError ()

4.15.1 Detailed Description

Definition at line 9 of file TypeError.h.

4.15.2 Constructor & Destructor Documentation

4.15.2.1 TypeError()

```
TypeError::TypeError ( )
```

Definition at line 3 of file TypeError.cpp.

```
    :std::runtime_error ("the number must be 1 or 0!"){};
```

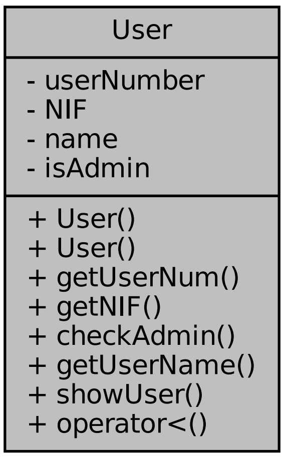
The documentation for this class was generated from the following files:

- TypeError.h
- TypeError.cpp

4.16 User Class Reference

```
#include <User.h>
```

Collaboration diagram for User:



Public Member Functions

- User ()
- User (float userNumber, float NIF, string name, bool isAdmin)
- int getUserNum ()
- int getNIF ()
- bool checkAdmin ()
- string getUserName ()
- void showUser ()
- bool operator< (const User &) const

Private Attributes

- int userNumber
- int NIF
- char name [10]
- bool isAdmin

4.16.1 Detailed Description

Definition at line 15 of file User.h.

4.16.2 Constructor & Destructor Documentation

4.16.2.1 User() [1/2]

```
User::User ( )
```

Definition at line 9 of file User.cpp.

```
{} //we define the default constructor
```

4.16.2.2 User() [2/2]

```
User::User (
              float userNumber,
              float NIF,
              string name,
              bool isAdmin )
```

Definition at line 11 of file User.cpp.

```
                                                                    { //we define the constructor with
       parameters
    this -> userNumber = userNumber;
    this -> NIF = NIF;
    strcpy(this->name,name.c_str());
    this -> isAdmin = isAdmin;
};
```

4.16.3 Member Function Documentation

4.16.3.1 checkAdmin()

`bool User::checkAdmin ( )`

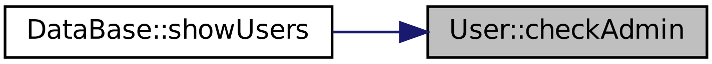
Returns if a user is admin o not by true or false

Definition at line 31 of file User.cpp.

```
                     {
  return isAdmin;
}
```

Referenced by DataBase::showUsers().

Here is the caller graph for this function:



4.16.3.2 getNIF()

`int User::getNIF ( )`

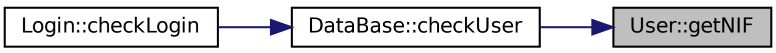
Returns the NIF

Definition at line 24 of file User.cpp.

```
                 {
  if (NIF < 9999999 || NIF > 99999999){
    throw NIFException();
  }
  return NIF;
}
```

Referenced by DataBase::checkUser().

Here is the caller graph for this function:



4.16.3.3 getUserName()

```
string User::getUserName ( )
```

Returns the user name

Definition at line 35 of file User.cpp.

```
                        {
  return name;
}
```

4.16.3.4 getUserNum()

```
int User::getUserNum ( )
```

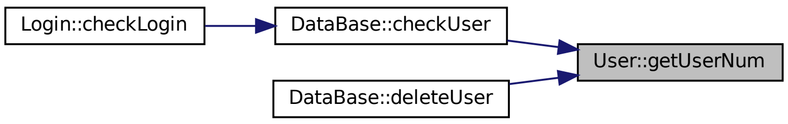
Returns the user number

Definition at line 18 of file User.cpp.

```
                     {
  if (userNumber < 1 || userNumber > 99999){
    throw UserNumException();
  }
  return userNumber;
}
```

Referenced by DataBase::checkUser(), and DataBase::deleteUser().

Here is the caller graph for this function:



4.16.3.5 operator<()

```
bool User::operator< (
             const User & user ) const
```

Overloads the < operator to compare in the set

Definition at line 43 of file User.cpp.

```
                                        {
  return userNumber < user.userNumber;
}
```

References userNumber.

4.16.3.6 showUser()

```
void User::showUser ( )
```

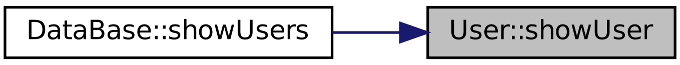
Returns the user name

Definition at line 39 of file User.cpp.

```
                      {
    cout « userNumber « "\t" « name;
}
```

Referenced by DataBase::showUsers().

Here is the caller graph for this function:



4.16.4 Member Data Documentation

4.16.4.1 isAdmin

```
bool User::isAdmin  [private]
```

Definition at line 56 of file User.h.

4.16.4.2 name

```
char User::name[10]  [private]
```

Definition at line 55 of file User.h.

4.16.4.3 NIF

```
int User::NIF  [private]
```

Definition at line 54 of file User.h.

4.16.4.4 userNumber

`int User::userNumber [private]`

Definition at line 53 of file User.h.

Referenced by operator<().

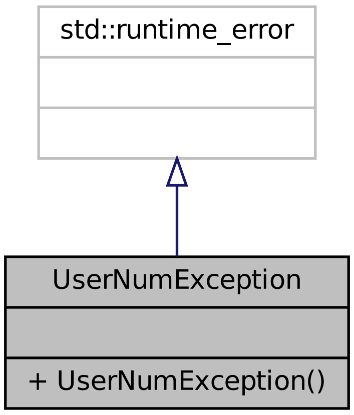
The documentation for this class was generated from the following files:

- User.h
- User.cpp

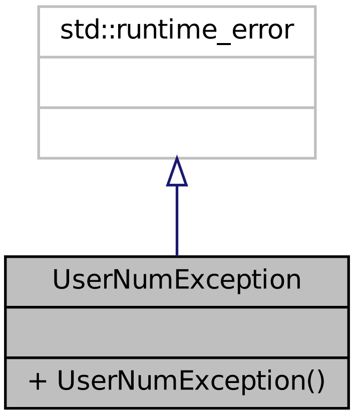
4.17 UserNumException Class Reference

`#include <UserNumException.h>`

Inheritance diagram for UserNumException:



Collaboration diagram for UserNumException:



Public Member Functions

- UserNumException ()

4.17.1 Detailed Description

Definition at line 9 of file UserNumException.h.

4.17.2 Constructor & Destructor Documentation

4.17.2.1 UserNumException()

```
UserNumException::UserNumException ( )
```

Definition at line 3 of file UserNumException.cpp.

```
4   :std::runtime_error ("the number of digits in user must be between 1 and 5."){};
```

The documentation for this class was generated from the following files:

- UserNumException.h
- UserNumException.cpp

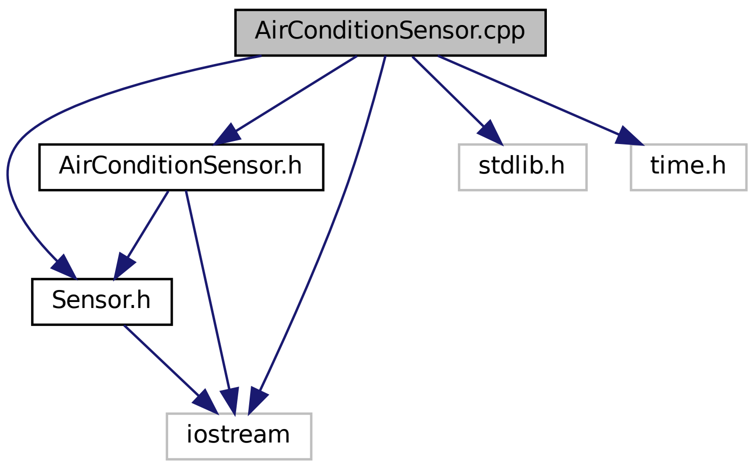
Chapter 5

File Documentation

5.1 AirConditionSensor.cpp File Reference

```
#include "AirConditionSensor.h"
#include "Sensor.h"
#include <iostream>
#include <stdlib.h>
#include <time.h>
```

Include dependency graph for AirConditionSensor.cpp:

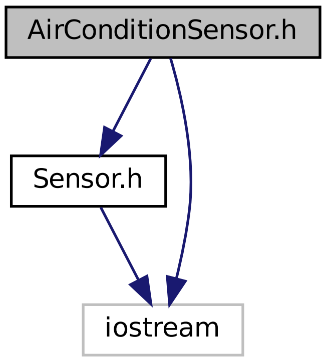


5.2 AirConditionSensor.h File Reference

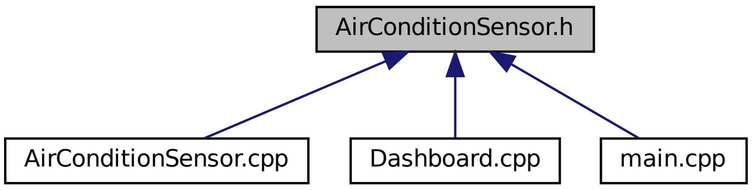
defines the AirConditionSensor class, wich is is inherited from the sensor class, with its attributes, methods, and constructor

```
#include "Sensor.h"
#include <iostream>
```

Include dependency graph for AirConditionSensor.h:



This graph shows which files directly or indirectly include this file:



Classes

- class AirConditionSensor

5.2.1 Detailed Description

defines the AirConditionSensor class, wich is is inherited from the sensor class, with its attributes, methods, and constructor

Author

Ana Martínez Albendea

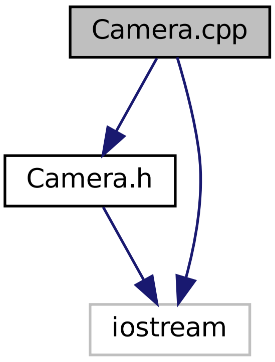
Date

2022-11-23

5.3 Camera.cpp File Reference

```
#include "Camera.h"
#include <iostream>
```

Include dependency graph for Camera.cpp:

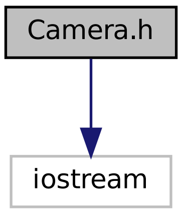


5.4 Camera.h File Reference

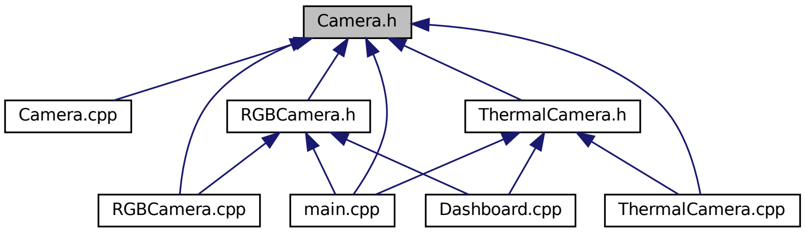
defines the camera class with its attributes, methods, and constructor

```
#include <iostream>
```

Include dependency graph for Camera.h:



This graph shows which files directly or indirectly include this file:



Classes

- class Camera

5.4.1 Detailed Description

defines the camera class with its attributes, methods, and constructor

Author

Ana Martínez Albendea

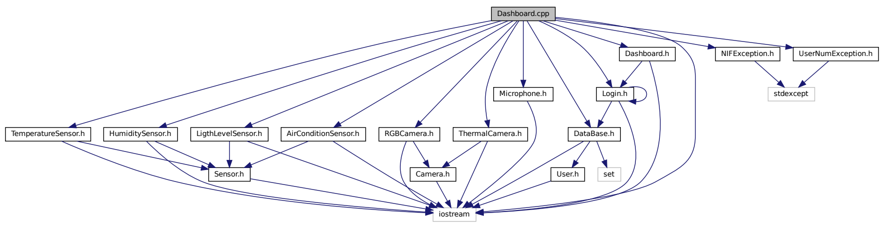
Date

2022-11-23

5.5 Dashboard.cpp File Reference

```
#include "TemperatureSensor.h"
#include "HumiditySensor.h"
#include "LigthLevelSensor.h"
#include "AirConditionSensor.h"
#include "RGBCamera.h"
#include "ThermalCamera.h"
#include "Microphone.h"
#include "DataBase.h"
#include "Login.h"
#include "Dashboard.h"
#include "UserNumException.h"
#include "NIFException.h"
#include <iostream>
```

Include dependency graph for Dashboard.cpp:

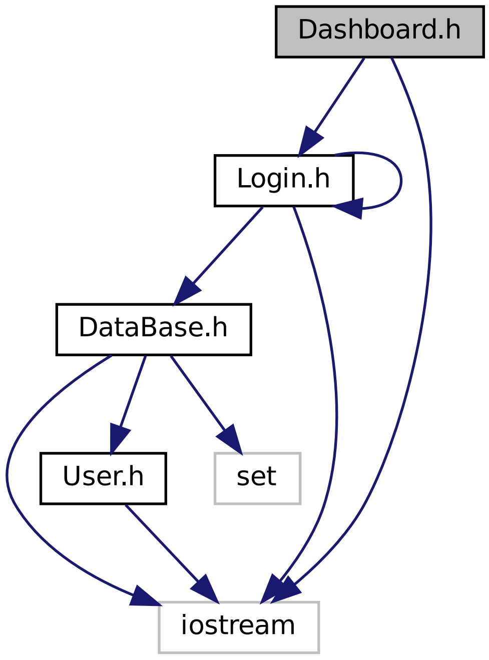


5.6 Dashboard.h File Reference

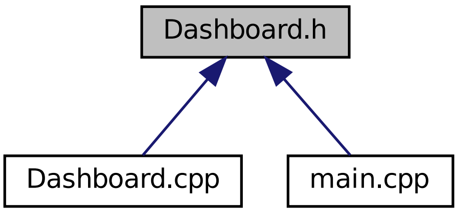
defines the dashboard class with its attributes, methods, and constructor

```
#include "Login.h"
#include <iostream>
```

Include dependency graph for Dashboard.h:



This graph shows which files directly or indirectly include this file:



Classes

- class Dashboard

5.6.1 Detailed Description

defines the dashboard class with its attributes, methods, and constructor

Author

Ana Martínez Albendea

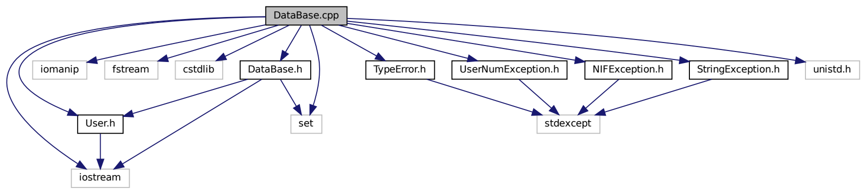
Date

2022-11-23

5.7 DataBase.cpp File Reference

```
#include "User.h"
#include <iomanip>
#include <fstream>
#include <cstdlib>
#include "DataBase.h"
#include "TypeError.h"
#include "UserNumException.h"
#include "NIFException.h"
#include "StringException.h"
#include <unistd.h>
#include <iostream>
#include <set>
```

Include dependency graph for DataBase.cpp:



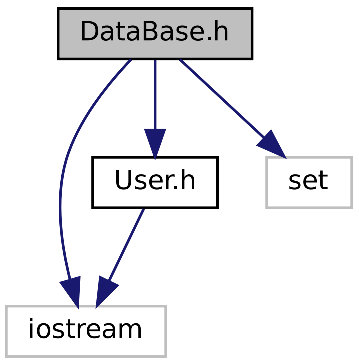
5.8 DataBase.h File Reference

defines the database class with its attributes, methods, and constructor

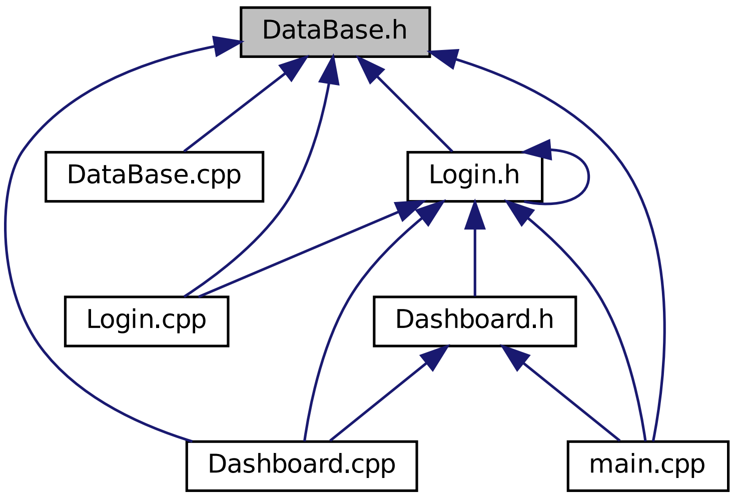
```
#include <iostream>
#include "User.h"
```

`#include <set>`

Include dependency graph for DataBase.h:



This graph shows which files directly or indirectly include this file:



Classes

- class DataBase

5.8.1 Detailed Description

defines the database class with its attributes, methods, and constructor

Author

Ana Martínez Albendea

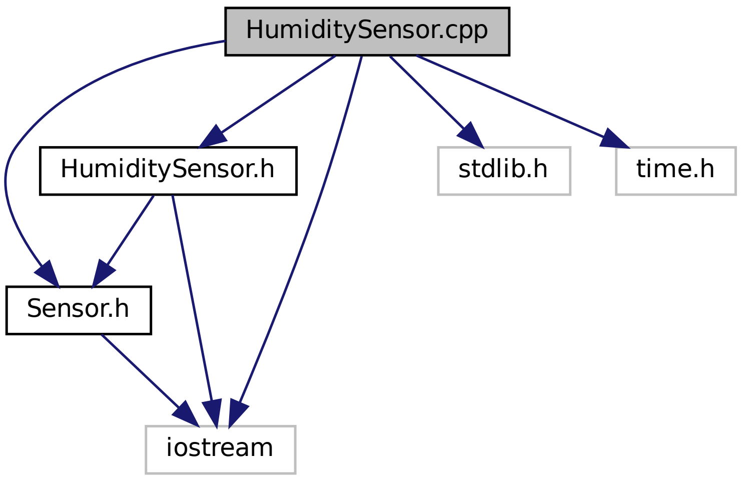
Date

2022-11-23

5.9 HumiditySensor.cpp File Reference

```
#include "HumiditySensor.h"
#include "Sensor.h"
#include <iostream>
#include <stdlib.h>
#include <time.h>
```

Include dependency graph for HumiditySensor.cpp:

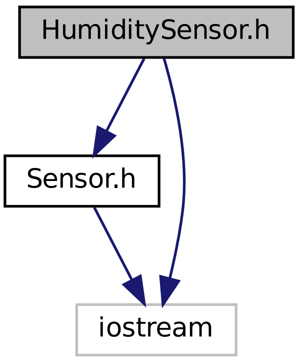


5.10 HumiditySensor.h File Reference

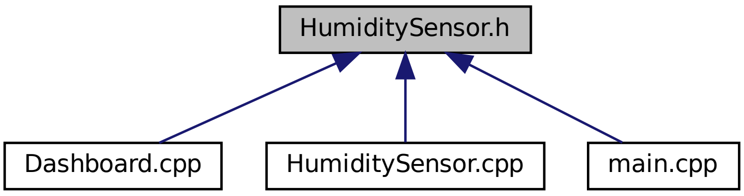
defines the HumiditySensor class, wich is is inherited from the sensor class, with its attributes, methods, and constructor

```
#include "Sensor.h"
#include <iostream>
```

Include dependency graph for HumiditySensor.h:



This graph shows which files directly or indirectly include this file:



Classes

- class HumiditySensor

5.10.1 Detailed Description

defines the HumiditySensor class, wich is is inherited from the sensor class, with its attributes, methods, and constructor

Author

Ana Martínez Albendea

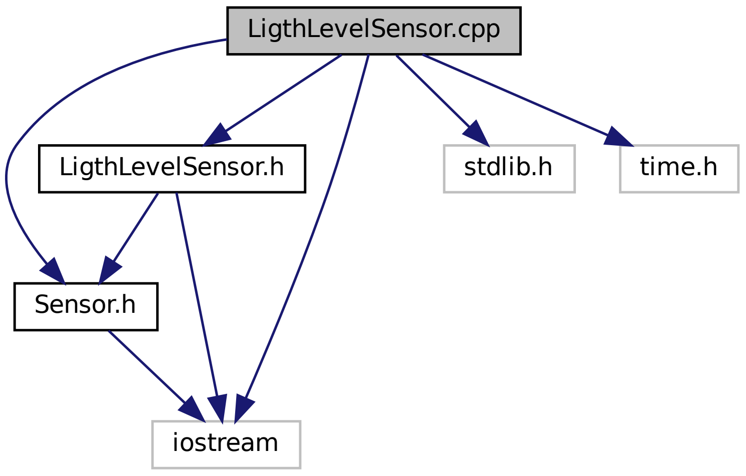
Date

2022-11-23

5.11 LigthLevelSensor.cpp File Reference

```
#include "LigthLevelSensor.h"
#include "Sensor.h"
#include <iostream>
#include <stdlib.h>
#include <time.h>
```

Include dependency graph for LigthLevelSensor.cpp:

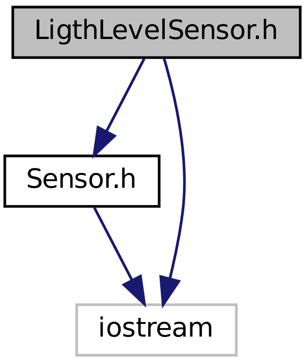


5.12 LigthLevelSensor.h File Reference

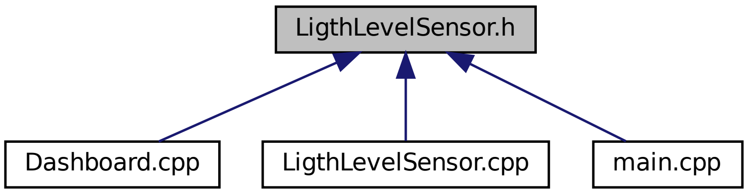
defines the LigthLevelSensor class, wich is is inherited from the sensor class, with its attributes, methods, and constructor

```
#include "Sensor.h"
#include <iostream>
```

Include dependency graph for LigthLevelSensor.h:



This graph shows which files directly or indirectly include this file:



Classes

- class LigthLevelSensor

5.12.1 Detailed Description

defines the LigthLevelSensor class, wich is is inherited from the sensor class, with its attributes, methods, and constructor

Author

Ana Martínez Albendea

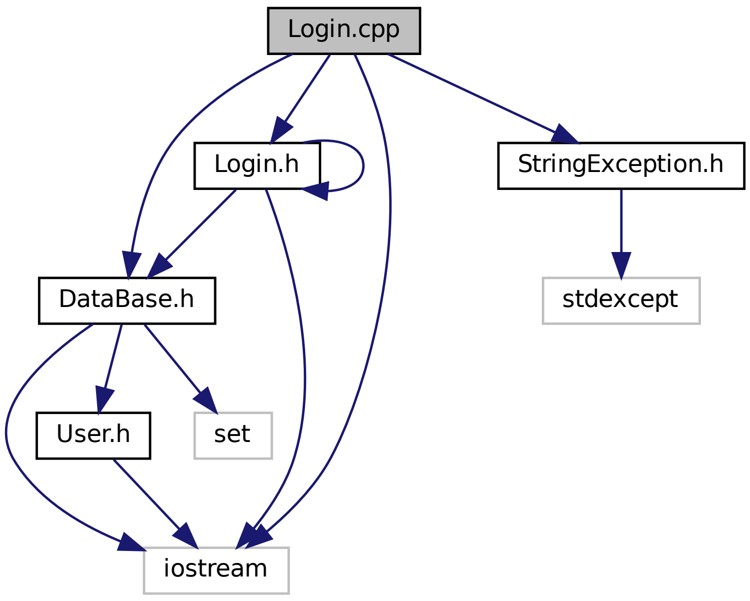
Date

2022-11-23

5.13 Login.cpp File Reference

```
#include "Login.h"
#include <iostream>
#include "StringException.h"
#include "DataBase.h"
```

Include dependency graph for Login.cpp:

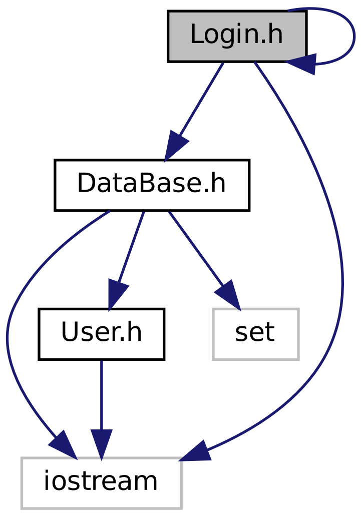


5.14 Login.h File Reference

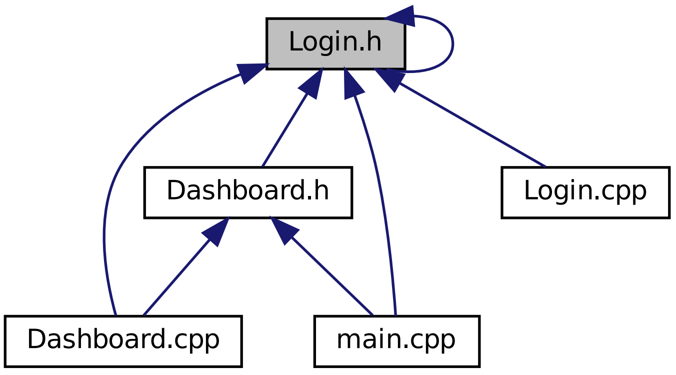
defines the login class with its attributes, methods, and constructor

```
#include "Login.h"
#include "DataBase.h"
#include <iostream>
```

Include dependency graph for Login.h:



This graph shows which files directly or indirectly include this file:



Classes

- class Login

5.14.1 Detailed Description

defines the login class with its attributes, methods, and constructor

Author

Ana Martínez Albendea

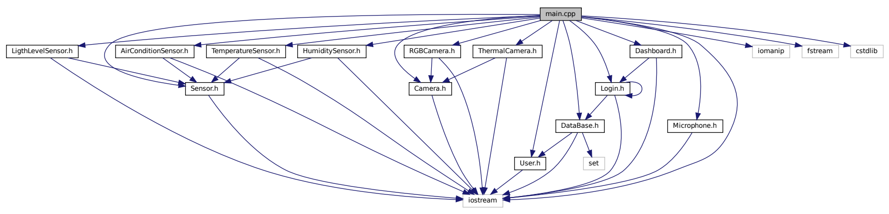
Date

2022-11-23

5.15 main.cpp File Reference

```
#include <iostream>
#include <iomanip>
#include <fstream>
#include <cstdlib>
#include "Sensor.h"
#include "TemperatureSensor.h"
#include "HumiditySensor.h"
#include "LigthLevelSensor.h"
#include "AirConditionSensor.h"
#include "Camera.h"
#include "RGBCamera.h"
#include "ThermalCamera.h"
#include "User.h"
#include "Microphone.h"
#include "DataBase.h"
#include "Login.h"
#include "Dashboard.h"
```

Include dependency graph for main.cpp:



Functions

- int main ()

5.15.1 Function Documentation

5.15.1.1 main()

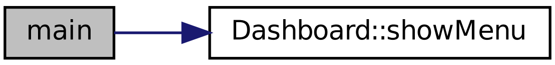
```
int main ( )
```

Definition at line 28 of file main.cpp.

```
28          {
29   //title created with ascii art
30   system("clear");
31   cout «
       "°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°" «
       endl;
32   cout «
       "------------------------------------------------------------------------------------------------" «
       endl;
33   cout « "      _      _ _         __     __                                        _   _ _   _
       " « endl
34   « "     | |_   _| (_) ___   \\ \\   / /__  __ _  __ _ _ __   ___   ___    ___  | | | (_)(_) ___   ___ " «
       endl
35   « "  _  | | | | | | |/ _ \\   \\ \\ / / _ \\/ _` |/ _` | '_ \\ / _ \\/ __|  / _ \\ | |_| | || |/ _ \\/
       __|" « endl
36   « " | |_| | |_| | | | (_) |   \\ V /  __/ (_| | (_| | | | | (_) \\__ \\ |  __/ |  _  | || | (_) \\__
       \\" « endl
37   « "  \\___/ \\__,_|_|_|\\___/     \\_/ \\___|\\__, |\\__,_|_| |_|\\___/|___/  \\___| |_| |_|_|/
       |\\___/|___/" « endl
38   « "                                   |___/                                    |__/           " «
       endl;
39   cout «
       "\n------------------------------------------------------------------------------------------------" «
       endl;
40   cout «
       "°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°°\n" «
       endl;
41
42
43   cout « "                              Please, log in to continue\n"  « endl;
44   try{
45
46     // Creates the binary file users.dat and save 10 spaces to store users. Can be created without that
       reservation
47     /*ofstream outUsersFile ("users.dat", ios::out | ios::binary);
48
49     if (!outUsersFile) { // ofstream could not open file
50       cerr « "File could not be opened." « endl;
51       exit (1);
52     }
53
54     User user; // fill with zeros each data member
55     for (int i = 0; i < 10; i++){ // write 10 empty records to file
56       outUsersFile.write (reinterpret_cast <const char *> (&user), sizeof (User));
57     }*/
58
59     Dashboard Ds; //construction of a dashboard type object
60     Dashboard* Dsptr = new Dashboard();
61
62     //while loop allowing to log back in after log out
63     while (true) {
64       Dsptr -> Dashboard::showMenu(); //call to the function that shows the main menu of the dashboard
65     }
66   }
67
68   catch(bad_alloc &except){
69     cout « "Exception: " « except.what() « endl;
70   }
71 }
```

References Dashboard::showMenu().

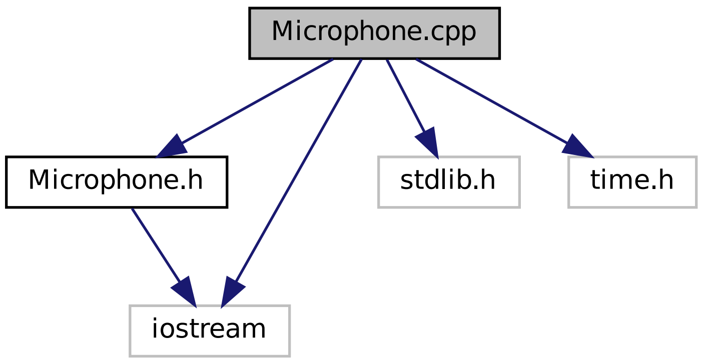
Here is the call graph for this function:



5.16 Microphone.cpp File Reference

```
#include "Microphone.h"
#include <iostream>
#include <stdlib.h>
#include <time.h>
```

Include dependency graph for Microphone.cpp:

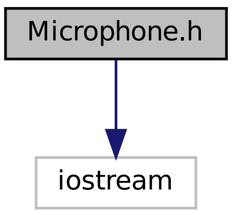


5.17 Microphone.h File Reference

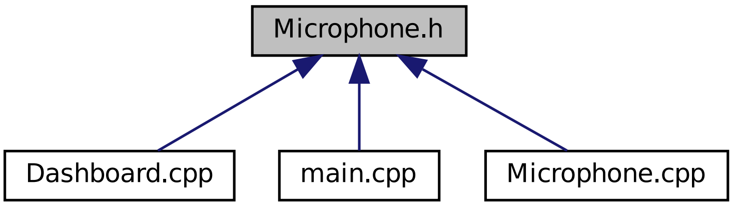
defines the microphone class with its attributes, methods, and constructor

```
#include <iostream>
```

Include dependency graph for Microphone.h:



This graph shows which files directly or indirectly include this file:



Classes

- class Microphone

5.17.1 Detailed Description

defines the microphone class with its attributes, methods, and constructor

Author

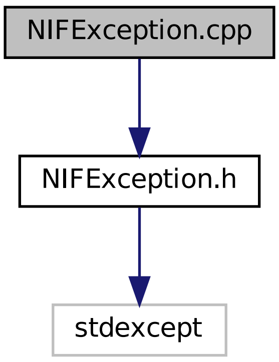
Ana Martínez Albendea

Date

2022-11-23

5.18 NIFException.cpp File Reference

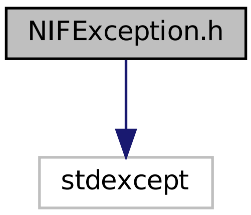
```
#include "NIFException.h"
```
Include dependency graph for NIFException.cpp:



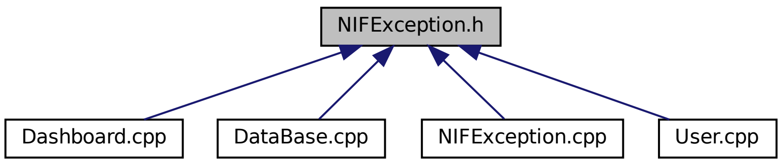
5.19 NIFException.h File Reference

defines the NIF exception class with its constructor

`#include <stdexcept>`
Include dependency graph for NIFException.h:



This graph shows which files directly or indirectly include this file:



Classes

- class NIFException

5.19.1 Detailed Description

defines the NIF exception class with its constructor

Author

Ana Martínez Albendea

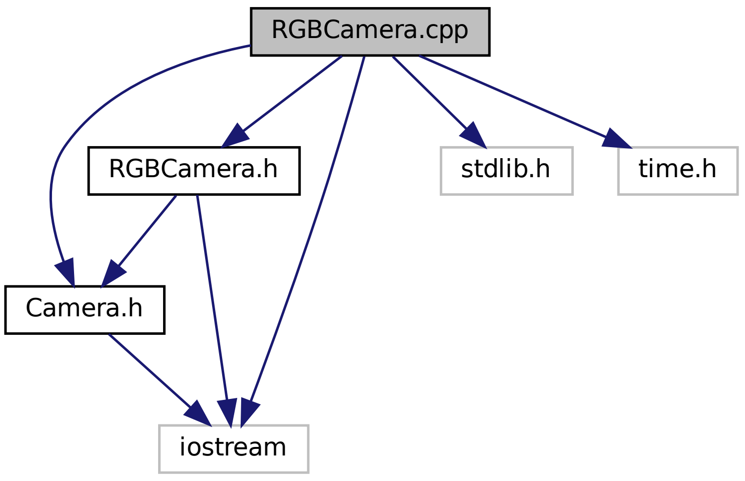
Date

2022-11-23

5.20 RGBCamera.cpp File Reference

```
#include "RGBCamera.h"
#include "Camera.h"
#include <iostream>
#include <stdlib.h>
#include <time.h>
```

Include dependency graph for RGBCamera.cpp:

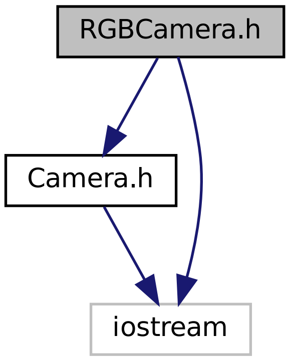


5.21 RGBCamera.h File Reference

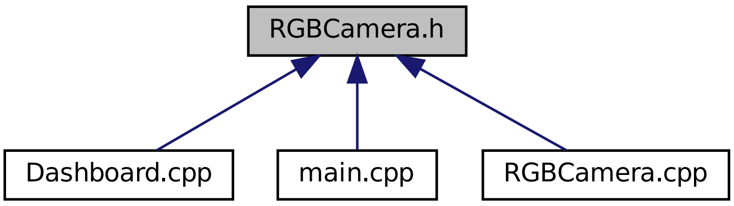
defines the RGBCamera class, wich is is inherited from the camera class, with its attributes, methods, and constructor

```
#include "Camera.h"
#include <iostream>
```

Include dependency graph for RGBCamera.h:



This graph shows which files directly or indirectly include this file:



Classes

- class RGBCamera

5.21.1 Detailed Description

defines the RGBCamera class, wich is is inherited from the camera class, with its attributes, methods, and constructor

Author

Ana Martínez Albendea

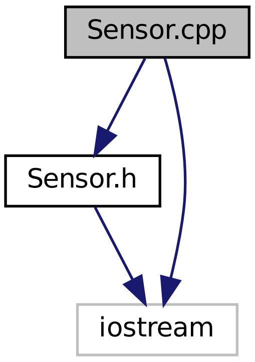
Date

2022-11-23

5.22 Sensor.cpp File Reference

```
#include "Sensor.h"
#include <iostream>
```

Include dependency graph for Sensor.cpp:

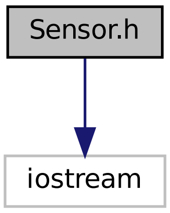


5.23 Sensor.h File Reference

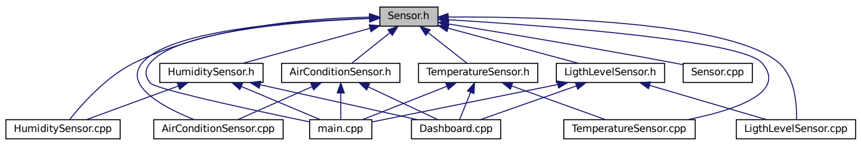
defines the sensor class with its attributes, methods, and constructor

```
#include <iostream>
```

Include dependency graph for Sensor.h:



This graph shows which files directly or indirectly include this file:



Classes

- class Sensor

5.23.1 Detailed Description

defines the sensor class with its attributes, methods, and constructor

Author

Ana Martínez Albendea

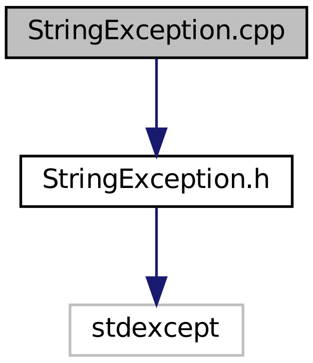
Date

2022-11-23

5.24 StringException.cpp File Reference

```
#include "StringException.h"
```

Include dependency graph for StringException.cpp:

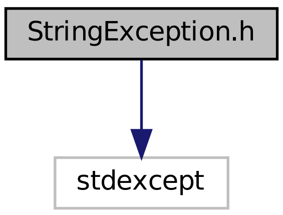


5.25 StringException.h File Reference

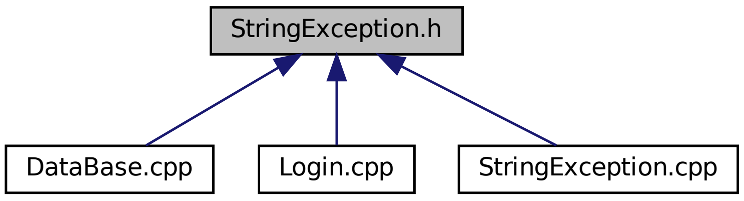
defines the string exception class with its constructor

```
#include <stdexcept>
```

Include dependency graph for StringException.h:



This graph shows which files directly or indirectly include this file:



Classes

- class StringException

5.25.1 Detailed Description

defines the string exception class with its constructor

Author

Ana Martínez Albendea

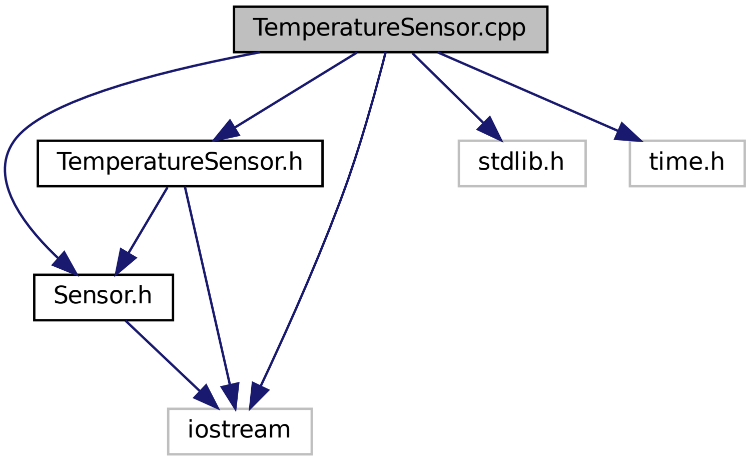
Date

2022-11-23

5.26 TemperatureSensor.cpp File Reference

```
#include "TemperatureSensor.h"
#include "Sensor.h"
#include <iostream>
#include <stdlib.h>
#include <time.h>
```

Include dependency graph for TemperatureSensor.cpp:

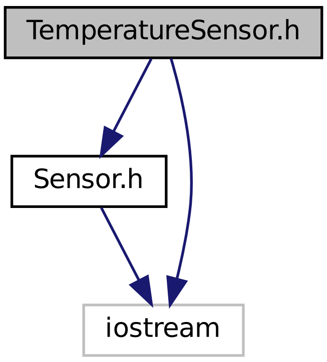


5.27 TemperatureSensor.h File Reference

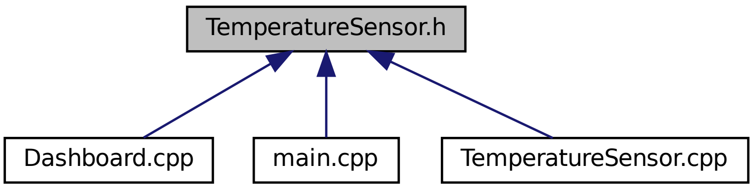
defines the TemperatureSensor class, wich is is inherited from the sensor class, with its attributes, methods, and constructor

```
#include "Sensor.h"
#include <iostream>
```

Include dependency graph for TemperatureSensor.h:



This graph shows which files directly or indirectly include this file:



Classes

- class TemperatureSensor

5.27.1 Detailed Description

defines the TemperatureSensor class, wich is is inherited from the sensor class, with its attributes, methods, and constructor

Author

Ana Martínez Albendea

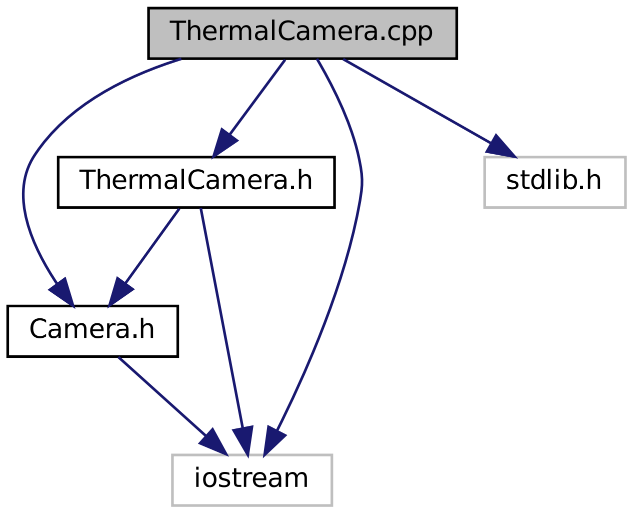
Date

2022-11-23

5.28 ThermalCamera.cpp File Reference

```
#include "ThermalCamera.h"
#include "Camera.h"
#include <iostream>
#include <stdlib.h>
```

Include dependency graph for ThermalCamera.cpp:

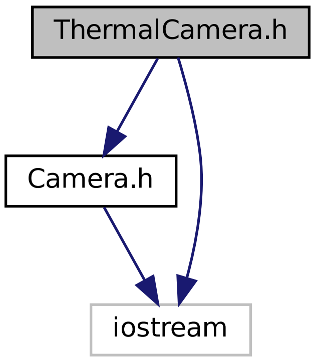


5.29 ThermalCamera.h File Reference

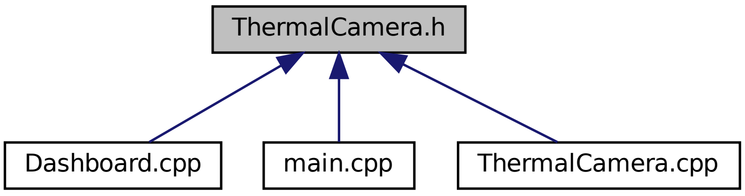
defines the ThermalCamera class, wich is is inherited from the camera class, with its attributes, methods, and constructor

```
#include "Camera.h"
#include <iostream>
```

Include dependency graph for ThermalCamera.h:



This graph shows which files directly or indirectly include this file:



Classes

- class ThermalCamera

5.29.1 Detailed Description

defines the ThermalCamera class, wich is is inherited from the camera class, with its attributes, methods, and constructor

Author

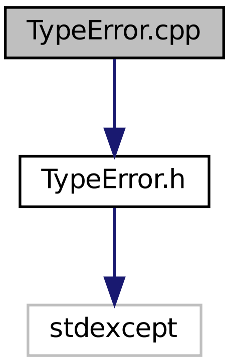
Ana Martínez Albendea

Date

2022-11-23

5.30 TypeError.cpp File Reference

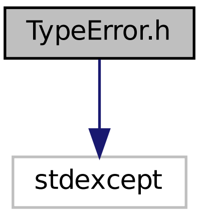
```
#include "TypeError.h"
```
Include dependency graph for TypeError.cpp:



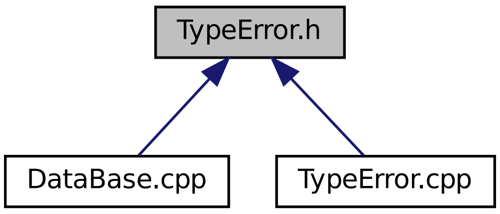
5.31 TypeError.h File Reference

defines the user type exception class with its constructor

```
#include <stdexcept>
```
Include dependency graph for TypeError.h:



This graph shows which files directly or indirectly include this file:



Classes

- class TypeError

5.31.1 Detailed Description

defines the user type exception class with its constructor

Author

Ana Martínez Albendea

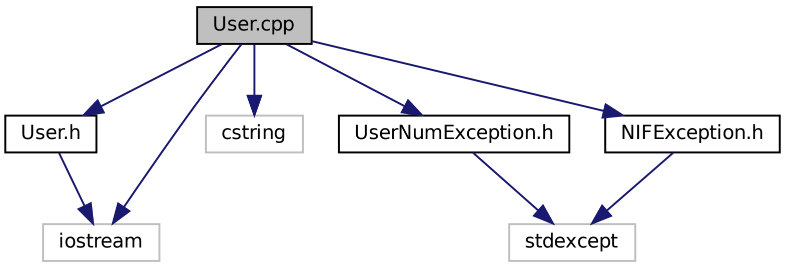
Date

2022-11-23

5.32 User.cpp File Reference

```
#include "User.h"
#include <iostream>
#include <cstring>
#include "UserNumException.h"
#include "NIFException.h"
```

Include dependency graph for User.cpp:

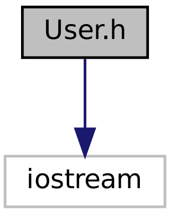


5.33 User.h File Reference

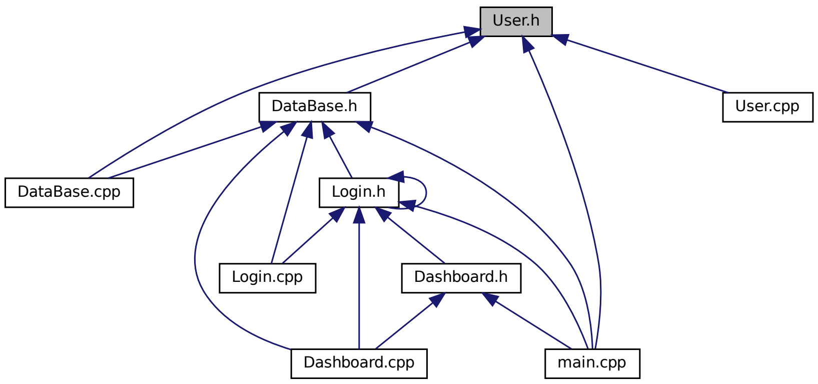
defines the user class with its attributes, methods, and constructor

```
#include <iostream>
```

Include dependency graph for User.h:



This graph shows which files directly or indirectly include this file:



Classes

- class User

5.33.1 Detailed Description

defines the user class with its attributes, methods, and constructor

Author

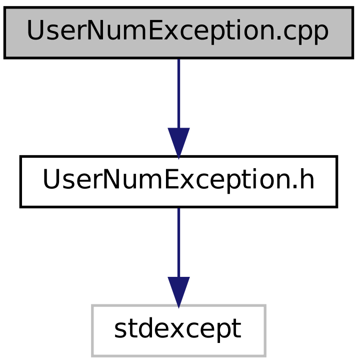
Ana Martínez Albendea

Date

2022-11-23

5.34 UserNumException.cpp File Reference

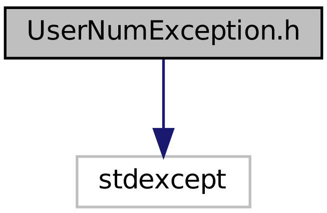
`#include "UserNumException.h"`
Include dependency graph for UserNumException.cpp:



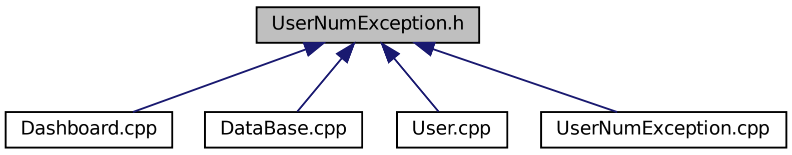
5.35 UserNumException.h File Reference

defines the user number exception class with its constructor

`#include <stdexcept>`
Include dependency graph for UserNumException.h:



This graph shows which files directly or indirectly include this file:



Classes

- class UserNumException

5.35.1 Detailed Description

defines the user number exception class with its constructor

Author

Ana Martínez Albendea

Date

2022-11-23

Index

A
    Dashboard, 18
addUser
    DataBase, 22
adminMenu
    Login, 40
airCondition
    AirConditionSensor, 11
AirConditionSensor, 7
    airCondition, 11
    AirConditionSensor, 8
    checkAirCondition, 9
    choice, 11
    getAirCondition, 9
    showChoice, 10
AirConditionSensor.cpp, 73
AirConditionSensor.h, 73

Camera, 12
    Camera, 13
    state, 15
    turnOff, 13
    turnOn, 14
Camera.cpp, 75
Camera.h, 75
checkAdmin
    User, 67
checkAirCondition
    AirConditionSensor, 9
checkHumidity
    HumiditySensor, 31
checkLigthLevel
    LigthLevelSensor, 36
checkLogin
    Login, 40
checkTemperature
    TemperatureSensor, 57
checkUser
    DataBase, 23
choice
    AirConditionSensor, 11
    DataBase, 27
    HumiditySensor, 33
    LigthLevelSensor, 38
    Microphone, 45
    RGBCamera, 50
    TemperatureSensor, 59
    ThermalCamera, 63

D
    Dashboard, 18
Dashboard, 15
    A, 18
    D, 18
    Dashboard, 16
    getOption, 16
    H, 18
    L, 19
    Li, 19
    M, 19
    option, 19
    RC, 19
    showMenu, 17
    T, 19
    TC, 20
Dashboard.cpp, 76
Dashboard.h, 77
DataBase, 20
    addUser, 22
    checkUser, 23
    choice, 27
    DataBase, 22
    dataBaseUser, 27
    deleteUser, 24
    isAdmin, 27
    it, 27
    name, 27
    NIF, 28
    NIFStr, 28
    saveFile, 25
    showAdminChoices, 25
    showUsers, 26
    type, 28
    userDeleted, 28
    userNow, 28
    userNumber, 28
    userNumberStr, 29
DataBase.cpp, 78
DataBase.h, 78
dataBaseUser
    DataBase, 27
deleteUser
    DataBase, 24

getAirCondition
    AirConditionSensor, 9
getHumidity
    HumiditySensor, 31
getLigthLevel
    LigthLevelSensor, 36

getNIF
  User, 67
getOption
  Dashboard, 16
getRGBImage
  RGBCamera, 49
getSound
  Microphone, 43
getTemp
  TemperatureSensor, 57
getThermalImage
  ThermalCamera, 62
getUserName
  User, 67
getUserNum
  User, 68

H
  Dashboard, 18
humidity
  HumiditySensor, 33
HumiditySensor, 29
  checkHumidity, 31
  choice, 33
  getHumidity, 31
  humidity, 33
  HumiditySensor, 30
  showChoice, 32
HumiditySensor.cpp, 80
HumiditySensor.h, 80

isAdmin
  DataBase, 27
  User, 69
it
  DataBase, 27

L
  Dashboard, 19
Li
  Dashboard, 19
ligthLevel
  LigthLevelSensor, 38
LigthLevelSensor, 34
  checkLigthLevel, 36
  choice, 38
  getLigthLevel, 36
  ligthLevel, 38
  LigthLevelSensor, 35
  showChoice, 37
LigthLevelSensor.cpp, 82
LigthLevelSensor.h, 82
Login, 39
  adminMenu, 40
  checkLogin, 40
  Login, 40
  NIF, 41
  NIFStr, 41
  userNow, 41
  userNowPtr, 41
  userNumber, 42
  userNumberStr, 42
Login.cpp, 84
Login.h, 84

M
  Dashboard, 19
main
  main.cpp, 86
main.cpp, 86
  main, 86
Microphone, 42
  choice, 45
  getSound, 43
  Microphone, 43
  showChoice, 43
  sound, 45
  state, 45
  turnOff, 44
  turnOn, 44
Microphone.cpp, 88
Microphone.h, 88

name
  DataBase, 27
  User, 69
NIF
  DataBase, 28
  Login, 41
  User, 69
NIFException, 46
  NIFException, 47
NIFException.cpp, 89
NIFException.h, 90
NIFStr
  DataBase, 28
  Login, 41

operator$<$
  User, 68
option
  Dashboard, 19

RC
  Dashboard, 19
RGBCamera, 47
  choice, 50
  getRGBImage, 49
  RGBCamera, 48
  showChoice, 49
RGBCamera.cpp, 91
RGBCamera.h, 91

saveFile
  DataBase, 25
Sensor, 51
  Sensor, 52
  state, 53

turnOff, 52
turnOn, 52
Sensor.cpp, 93
Sensor.h, 93
showAdminChoices
DataBase, 25
showChoice
AirConditionSensor, 10
HumiditySensor, 32
LigthLevelSensor, 37
Microphone, 43
RGBCamera, 49
TemperatureSensor, 58
ThermalCamera, 62
showMenu
Dashboard, 17
showUser
User, 68
showUsers
DataBase, 26
sound
Microphone, 45
state
Camera, 15
Microphone, 45
Sensor, 53
StringException, 54
StringException, 55
StringException.cpp, 94
StringException.h, 94

T
Dashboard, 19
TC
Dashboard, 20
temperature
TemperatureSensor, 59
TemperatureSensor, 55
checkTemperature, 57
choice, 59
getTemp, 57
showChoice, 58
temperature, 59
TemperatureSensor, 56
TemperatureSensor.cpp, 96
TemperatureSensor.h, 96
ThermalCamera, 60
choice, 63
getThermalImage, 62
showChoice, 62
ThermalCamera, 61
ThermalCamera.cpp, 98
ThermalCamera.h, 98
turnOff
Camera, 13
Microphone, 44
Sensor, 52
turnOn
Camera, 14
Microphone, 44
Sensor, 52
type
DataBase, 28
TypeError, 64
TypeError, 65
TypeError.cpp, 100
TypeError.h, 100

User, 65
checkAdmin, 67
getNIF, 67
getUserName, 67
getUserNum, 68
isAdmin, 69
name, 69
NIF, 69
operator<, 68
showUser, 68
User, 66
userNumber, 69
User.cpp, 101
User.h, 102
userDeleted
DataBase, 28
userNow
DataBase, 28
Login, 41
userNowPtr
Login, 41
userNumber
DataBase, 28
Login, 42
User, 69
userNumberStr
DataBase, 29
Login, 42
UserNumException, 70
UserNumException, 71
UserNumException.cpp, 103
UserNumException.h, 103