

## Practice 4

Generated by Doxygen 1.8.17



<b>1 Hierarchical Index</b>	<b>1</b>
1.1 Class Hierarchy	1
<b>2 Class Index</b>	<b>3</b>
2.1 Class List	3
<b>3 Class Documentation</b>	<b>5</b>
3.1 Admin Class Reference	5
3.2 CLDashboard Class Reference	6
3.2.1 Member Function Documentation	7
3.2.1.1 changeMainMenu()	7
3.2.1.2 errorCommand()	7
3.2.1.3 helpCommand()	8
3.2.1.4 listSensor()	8
3.2.1.5 permissionError()	8
3.2.1.6 readCommand()	8
3.2.1.7 showMainMenu()	8
3.3 CLInterface Class Reference	9
3.4 CLLoginInterface Class Reference	10
3.5 CLMenu Class Reference	11
3.6 CLMenuBar Class Reference	12
3.7 Dashboard Class Reference	13
3.7.1 Member Function Documentation	14
3.7.1.1 addNewSensor()	14
3.7.1.2 addToMainMenu() [1/2]	15
3.7.1.3 addToMainMenu() [2/2]	15
3.7.1.4 canExit()	15
3.7.1.5 changeCurrentSensorInfo()	16
3.7.1.6 changeInterface()	16
3.7.1.7 cleanSensor()	16
3.7.1.8 Create()	16
3.7.1.9 deleteSensor()	17
3.7.1.10 exit()	17
3.7.1.11 logout()	17
3.7.1.12 moveWindowMainMenu()	18
3.7.1.13 setUser()	18
3.8 Database Class Reference	18
3.9 Interface Class Reference	19
3.10 LoginInterface Class Reference	20
3.11 Menu Class Reference	21
3.12 MenuBar Class Reference	22
3.13 Sensor Class Reference	22
3.14 User Class Reference	23



# Chapter 1

## Hierarchical Index

### 1.1 Class Hierarchy

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

Dashboard . . . . .	13
CLDashboard . . . . .	6
Database . . . . .	18
Interface . . . . .	19
CLInterface . . . . .	9
LoginInterface . . . . .	20
CLLoginInterface . . . . .	10
Menu . . . . .	21
CLMenu . . . . .	11
MenuBar . . . . .	22
CLMenuBar . . . . .	12
Sensor . . . . .	22
User . . . . .	23
Admin . . . . .	5



## Chapter 2

# Class Index

### 2.1 Class List

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

<a href="#">Admin</a>	5
<a href="#">CLDashboard</a>	6
<a href="#">CLInterface</a>	9
<a href="#">CLLoginInterface</a>	10
<a href="#">CLMenu</a>	11
<a href="#">CLMenuBar</a>	12
<a href="#">Dashboard</a>	13
<a href="#">Database</a>	18
<a href="#">Interface</a>	19
<a href="#">LoginInterface</a>	20
<a href="#">Menu</a>	21
<a href="#">MenuBar</a>	22
<a href="#">Sensor</a>	22
<a href="#">User</a>	23



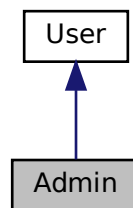


## Chapter 3

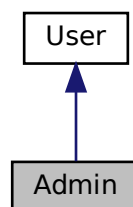
# Class Documentation

### 3.1 Admin Class Reference

Inheritance diagram for Admin:



Collaboration diagram for Admin:



### Public Member Functions

- **Admin** (std::string number="00000", std::string nif="00000000", std::string name="default")

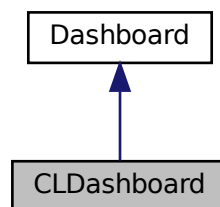
## Additional Inherited Members

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

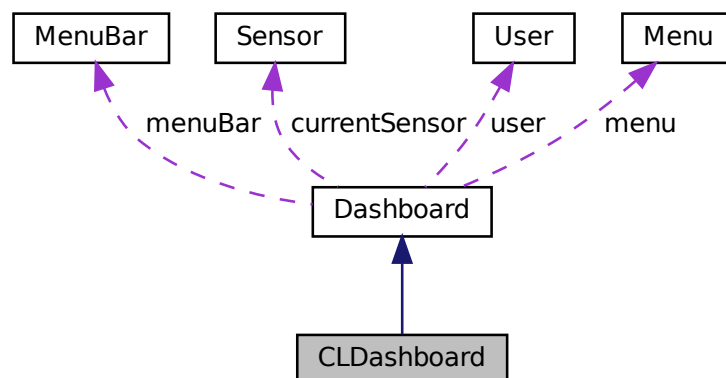
- include/Admin.h
- src/Admin.cpp

## 3.2 CLDashboard Class Reference

Inheritance diagram for CLDashboard:



Collaboration diagram for CLDashboard:



## Public Member Functions

- void [showMainMenu](#) ()  
*Shows the default menu and starts the custom terminal.*

## Protected Member Functions

- void `changeMainMenu` (int n)  
*Change the page of the menu.*
- void `readCommand` ()  
*Reads and executes a command from the terminal.*
- void `listSensor` ()  
*Command that lists the sensors in the current menu page.*
- void `helpCommand` (std::string command="")  
*Displays the use of the command.*
- void `errorCommand` (std::string command="")  
*Shows error message because the command not being found.*
- void `permissionError` ()  
*Shows error message because the permission being denied to execute it.*

## Additional Inherited Members

### 3.2.1 Member Function Documentation

#### 3.2.1.1 `changeMainMenu()`

```
void CLDashboard::changeMainMenu (  
    int n ) [protected]
```

Change the page of the menu.

##### Parameters

<i>n</i>	Number of pages to shift ( positive to the right )
----------	--

#### 3.2.1.2 `errorCommand()`

```
void CLDashboard::errorCommand (  
    std::string command = "" ) [protected]
```

Shows error message because the command not being found.

##### Parameters

<i>command</i>	Command input
----------------	---------------

### 3.2.1.3 helpCommand()

```
void CLDashboard::helpCommand (
    std::string command = "" ) [protected]
```

Displays the use of the command.

#### Parameters

<i>command</i>	Command to display info about
----------------	-------------------------------

### 3.2.1.4 listSensor()

```
void CLDashboard::listSensor ( ) [protected]
```

Command that lists the sensors in the current menu page.

### 3.2.1.5 permissionError()

```
void CLDashboard::permissionError ( ) [protected]
```

Shows error message because the permission being denied to execute it.

### 3.2.1.6 readCommand()

```
void CLDashboard::readCommand ( ) [protected]
```

Reads and executes a command from the terminal.

### 3.2.1.7 showMainMenu()

```
void CLDashboard::showMainMenu ( ) [virtual]
```

Shows the default menu and starts the custom terminal.

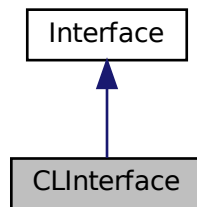
Implements [Dashboard](#).

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

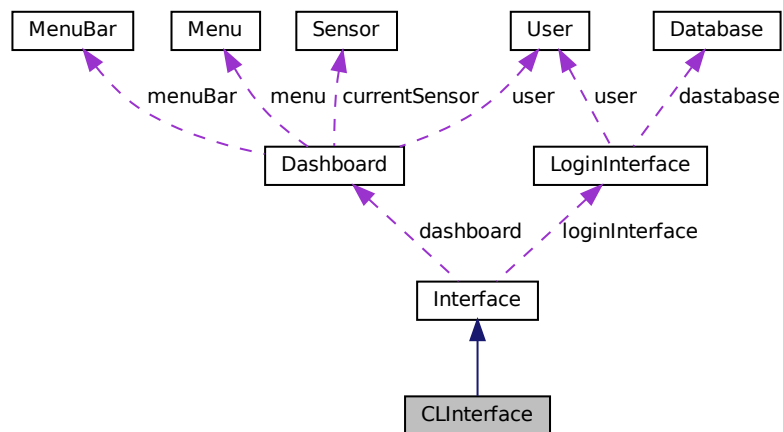
- include/CLDashboard.h
- src/CLDashboard.cpp

### 3.3 CLInterface Class Reference

Inheritance diagram for CLInterface:



Collaboration diagram for CLInterface:



#### Public Member Functions

- void **login** (int tries)
- void **loadMenu** ()

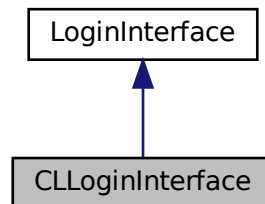
#### Additional Inherited Members

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

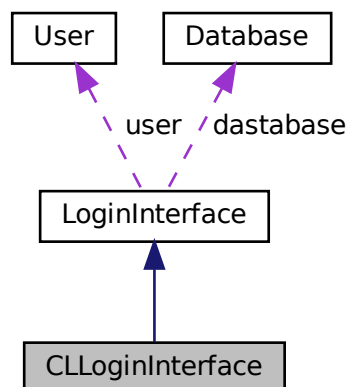
- include/CLInterface.h
- src/CLInterface.cpp

### 3.4 CLLoginInterface Class Reference

Inheritance diagram for CLLoginInterface:



Collaboration diagram for CLLoginInterface:



#### Public Member Functions

- void **showWelcomeMessage** ()
- void **askEmployeeNumber** ()
- void **askEmployeeNumber** (const std::string &userNumber)
- void **askNIF** ()
- void **askNIF** (const std::string &nif)

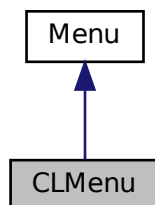
#### Additional Inherited Members

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

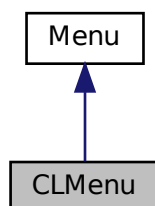
- include/CLLoginInterface.h
- src/CLLoginInterface.cpp

## 3.5 CLMenu Class Reference

Inheritance diagram for CLMenu:



Collaboration diagram for CLMenu:



### Public Member Functions

- void **show** (std::vector< [Sensor](#) \* >)
- void **show** ([Sensor](#) \*sensor)

### Protected Member Functions

- void **clearMenu** ()

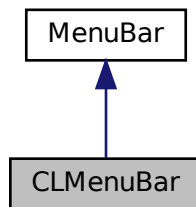
### Additional Inherited Members

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

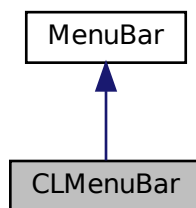
- include/CLMenu.h
- src/CLMenu.cpp

## 3.6 CLMenuBar Class Reference

Inheritance diagram for CLMenuBar:



Collaboration diagram for CLMenuBar:



### Public Member Functions

- void **show** ()

### Additional Inherited Members

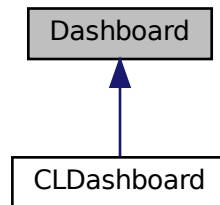
The documentation for this class was generated from the following files:

- include/CLMenuBar.h
- src/CLMenuBar.cpp

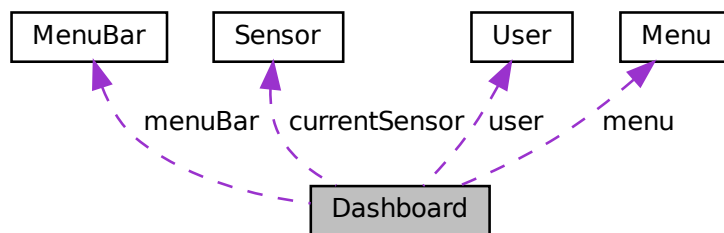


## 3.7 Dashboard Class Reference

Inheritance diagram for Dashboard:



Collaboration diagram for Dashboard:



### Public Member Functions

- virtual void **showMainMenu** ()=0
- void **changeInterface** (std::string newInterface="")  
Changes the current interface to the input one.
- void **addToMainMenu** ()  
Adds all the current sensors to the dashboard.
- void **addToMainMenu** (Sensor \*sensor)  
Adds a sensor to the dashboard.
- void **moveWindowMainMenu** (int n=1)  
Shifts the main menu n pages.
- void **changeCurrentSensorInfo** (std::string toChange, std::string newValue)  
Changes the current sensor values.
- void **setUser** (User user)  
Adds the user to the dashboard.
- void **addNewSensor** (std::string id)

- Creates the new sensor and calls addToMainMenu.*

  - void `deleteSensor` (std::string id)

*Deletes the desired sensor from the dashboard.*
- void `cleanSensor` ()

*Deletes all the sensors from the dashboard.*
- void `exit` ()

*Exits the dashboard and finishes the program.*
- void `logout` ()

*Exits the dashboard and goes back to the login screen.*
- bool `canExit` ()

*Called from main, checks if the program is allowed to exit.*

## Static Public Member Functions

- static `Dashboard * Create` (const std::string type="CLI")  
*Create a new instance of the desired `Dashboard` interface.*

## Protected Attributes

- std::vector< `Sensor` \* > `sensor`  
*Vector of all the vectors in the dashboard.*
- std::vector< std::vector< `Sensor` \* > > `mainMenu`  
*Vetor that contains the pages of the main `Menu`.*
- int `mainMenuIndex`  
*Current page of the main `Menu`.*
- `MenuBar` \* `menuBar`
- `Menu` \* `menu`
- `User` `user`
- std::string `currentInterface`  
*Current interface, can be a sensor ID or the main menu (..)*
- std::string `lastInterface`  
*Last interface, can be a sensor ID or the main menu (..)*
- `Sensor` \* `currentSensor`  
*If the currentInterface is a sensor then the value would be that sensor, if not the value is nullptr.*
- bool `allowedToExit`

## 3.7.1 Member Function Documentation

### 3.7.1.1 addNewSensor()

```
void Dashboard::addNewSensor (
    std::string id )
```

Creates the new sensor and calls addToMainMenu.

## Parameters

<i>id</i>	New sensor type
-----------	-----------------

## See also

[addToMainMenu](#)**3.7.1.2 addToMainMenu()** [1/2]

```
void Dashboard::addToMainMenu ( )
```

Adds all the current sensors to the dashboard.

## Note

Only called at the start of the program

**3.7.1.3 addToMainMenu()** [2/2]

```
void Dashboard::addToMainMenu (
    Sensor * sensor )
```

Adds a sensor to the dashboard.

## Parameters

<i>sensor</i>	<a href="#">Sensor</a> to add
---------------	-------------------------------

**3.7.1.4 canExit()**

```
bool Dashboard::canExit ( )
```

Called from main, checks if the program is allowed to exit.

## Returns

true End the program  
false Cannot exit, goes to login screen

### 3.7.1.5 `changeCurrentSensorInfo()`

```
void Dashboard::changeCurrentSensorInfo (
    std::string toChange,
    std::string newValue )
```

Changes the current sensor values.

#### Parameters

<i>toChange</i>	Desired value to change
<i>newValue</i>	New value to add

### 3.7.1.6 `changeInterface()`

```
void Dashboard::changeInterface (
    std::string newInterface = "" )
```

Changes the current interface to the input one.

#### Parameters

<i>newInterface</i>	Desired interface to change ( .. = main <a href="#">Menu</a> )
---------------------	--

### 3.7.1.7 `cleanSensor()`

```
void Dashboard::cleanSensor ( )
```

Deletes all the sensors from the dashboard.

#### Note

Only called when exiting the dashboard

### 3.7.1.8 `Create()`

```
Dashboard * Dashboard::Create (
    const std::string type = "CLI" ) [static]
```

Create a new instance of the desired [Dashboard](#) interface.

## Parameters

<i>type</i>	Type of <a href="#">Dashboard</a> interface
-------------	---

## Returns

Dashboard\*

**3.7.1.9 deleteSensor()**

```
void Dashboard::deleteSensor (
    std::string id )
```

Deletes the desired sensor from the dashboard.

## Parameters

<i>id</i>	ID of the sensor to delete
-----------	----------------------------

## See also

[addToMainMenu](#)

**3.7.1.10 exit()**

```
void Dashboard::exit ( )
```

Exits the dashboard and finishes the program.

## See also

[cleanSensor](#)

**3.7.1.11 logout()**

```
void Dashboard::logout ( )
```

Exits the dashboard and goes back to the login screen.

## See also

[cleanSensor](#)

### 3.7.1.12 moveWindowMainMenu()

```
void Dashboard::moveWindowMainMenu (
    int n = 1 )
```

Shifts the main menu n pages.

#### Parameters

<i>n</i>	Number of pages to shift ( positive to the right )
----------	--

### 3.7.1.13 setUser()

```
void Dashboard::setUser (
    User user )
```

Adds the user to the dashboard.

#### Parameters

<i>user</i>	Current user
-------------	--------------

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

- include/Dashboard.h
- src/Dashboard.cpp

## 3.8 Database Class Reference

### Public Member Functions

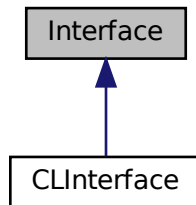
- void **addUser** ([User](#))
- [User](#) **getUser** (std::string, std::string)

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

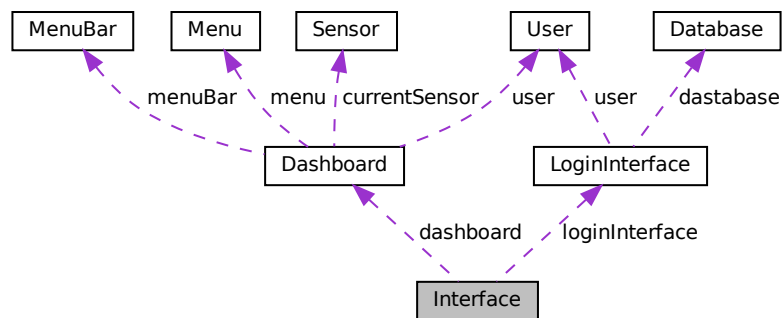
- include/Database.h
- src/Database.cpp

## 3.9 Interface Class Reference

Inheritance diagram for Interface:



Collaboration diagram for Interface:



### Public Member Functions

- virtual void **login** (int tries)=0
- void **login** (std::string userNumber, std::string nif)
- bool **loadMenu** ()

### Static Public Member Functions

- static [Interface](#) \* **Create** (const std::string type="CLI")

### Protected Attributes

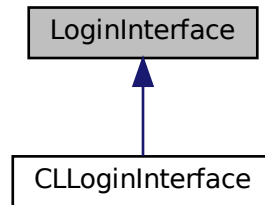
- [LoginInterface](#) \* **loginInterface**
- [Dashboard](#) \* **dashboard**

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

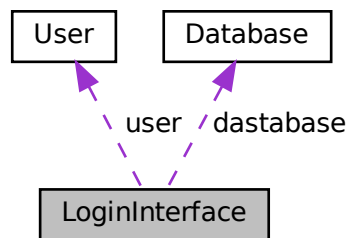
- include/Interface.h
- src/Interface.cpp

### 3.10 LoginInterface Class Reference

Inheritance diagram for LoginInterface:



Collaboration diagram for LoginInterface:



#### Public Member Functions

- virtual void **showWelcomeMessage** ()=0
- virtual void **askEmployeeNumber** ()=0
- virtual void **askEmployeeNumber** (const std::string &userNumber)=0
- virtual void **askNIF** ()=0
- virtual void **askNIF** (const std::string &userNumber)=0
- bool **checkUser** ()
- [User](#) **getUser** ()

#### Static Public Member Functions

- static [LoginInterface](#) \* **Create** (const std::string="CLI")



## Protected Attributes

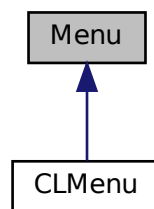
- `std::string` **inputNIF**
- `std::string` **inputEmployeeNumber**
- [User](#) **user**
- [Database](#) **dastabase**

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

- `include/LoginInterface.h`
- `src/LoginInterface.cpp`

## 3.11 Menu Class Reference

Inheritance diagram for Menu:



## Public Member Functions

- virtual void **show** (`std::vector< Sensor *` `>`)=0
- virtual void **show** (`Sensor *`)=0

## Static Public Member Functions

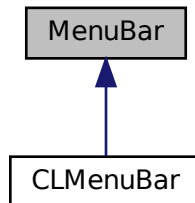
- static [Menu](#) \* **Create** (const `std::string` type="CLI")

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

- `include/Menu.h`
- `src/Menu.cpp`

## 3.12 MenuBar Class Reference

Inheritance diagram for MenuBar:



### Public Member Functions

- virtual void **show** ()=0
- void **setUserName** (std::string userName)
- void **setCurrentMenu** (std::string currentMenu)

### Static Public Member Functions

- static [MenuBar](#) \* **Create** (const std::string type="CLI")

### Protected Attributes

- std::string **userName**
- std::string **currentMenu**

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

- include/MenuBar.h
- src/MenuBar.cpp

## 3.13 Sensor Class Reference

### Public Member Functions

- **Sensor** (std::string id="0000", std::string type="none", std::string magnitude="-", bool active=true, int val↔ PerMin=1)
- std::string **getId** ()
- void **setActive** (bool)
- bool **isActive** ()
- std::string **getType** ()

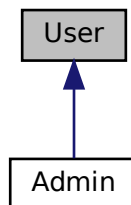
- void **setArea** (std::string)
- std::string **getArea** ()
- std::vector< int > **requestData** ()
- void **setMagnitude** (std::string magnitude)
- std::string **getMagnitude** ()
- void **setValPerMin** (int valPerMin)
- int **getValPerMin** ()

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

- include/Sensor.h
- src/Sensor.cpp

## 3.14 User Class Reference

Inheritance diagram for User:



### Public Member Functions

- **User** (std::string number="00000", std::string nif="00000000", std::string name="default")
- bool **isSameNIF** (std::string)
- bool **isSameEmployeeNumber** (std::string)
- std::string **getName** ()
- void **addTimestamp** ()
- std::string **getTimestamp** ()
- bool **hasAdminPermission** ()

### Protected Member Functions

- void **setNIF** (std::string)
- void **setEmployeeNumber** (std::string)

## Protected Attributes

- `std::string` **name**
- `std::string` **NIF**
- `std::string` **employeeNumber**
- `std::string` **lastLogTime**
- `bool` **adminPermission**

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

- `include/User.h`
- `src/User.cpp`

# Index

- addNewSensor
  - Dashboard, [14](#)
- addToMainMenu
  - Dashboard, [15](#)
- Admin, [5](#)
- canExit
  - Dashboard, [15](#)
- changeCurrentSensorInfo
  - Dashboard, [15](#)
- changeInterface
  - Dashboard, [16](#)
- changeMainMenu
  - CLDashboard, [7](#)
- CLDashboard, [6](#)
  - changeMainMenu, [7](#)
  - errorCommand, [7](#)
  - helpCommand, [7](#)
  - listSensor, [8](#)
  - permissionError, [8](#)
  - readCommand, [8](#)
  - showMainMenu, [8](#)
- cleanSensor
  - Dashboard, [16](#)
- CLInterface, [9](#)
- CLLoginInterface, [10](#)
- CLMenu, [11](#)
- CLMenuBar, [12](#)
- Create
  - Dashboard, [16](#)
- Dashboard, [13](#)
  - addNewSensor, [14](#)
  - addToMainMenu, [15](#)
  - canExit, [15](#)
  - changeCurrentSensorInfo, [15](#)
  - changeInterface, [16](#)
  - cleanSensor, [16](#)
  - Create, [16](#)
  - deleteSensor, [17](#)
  - exit, [17](#)
  - logout, [17](#)
  - moveWindowMainMenu, [17](#)
  - setUser, [18](#)
- Database, [18](#)
- deleteSensor
  - Dashboard, [17](#)
- errorCommand
  - CLDashboard, [7](#)
- exit
  - Dashboard, [17](#)
- helpCommand
  - CLDashboard, [7](#)
- Interface, [19](#)
- listSensor
  - CLDashboard, [8](#)
- LoginInterface, [20](#)
- logout
  - Dashboard, [17](#)
- Menu, [21](#)
- MenuBar, [22](#)
- moveWindowMainMenu
  - Dashboard, [17](#)
- permissionError
  - CLDashboard, [8](#)
- readCommand
  - CLDashboard, [8](#)
- Sensor, [22](#)
- setUser
  - Dashboard, [18](#)
- showMainMenu
  - CLDashboard, [8](#)
- User, [23](#)