# Tables

## Families

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Notes** |
| **tag** | Genetate by an hash function from “name”  PRIMARY KEY, varchar[12] | Control collision |
| **name** | Varchar[50], family name |  |

## Users

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Notes** |
| **id** | User id, primary key |  |
| **family\_tag** | Foreign key, relationship whit Families table |  |
| **user\_name** | Varchar[50] | This is NOT a UNIQUE KEY.  Control users with the same name only if they belong to the same family |
| **password** | Varchar[50] |  |
| **User\_type** | Foreign key, relationship whit User\_definition table | Default is “1” |

## Users\_definition

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Notes** |
| **id** | primary key |  |
| **name** | Varchar[50] |  |
| **description** | Varchar[150] |  |
| **Access\_level** | int | The lower the number, the deeper is the access permission |

## THINGS

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Notes** |
| **tag** | primary key, varchar[12] | This code identifies the device which send metrics to server |
| **name** | Varchar[50] |  |
| **family\_tag** | Foreign key, relationship whit Families table |  |

# PHP components

## Database()

|  |  |  |
| --- | --- | --- |
| **Method** | **Parameters** | **Description** |
| **\_\_constructor()** | void | Initialize private variable $conn |
| **closeConnection()** | void | Close connection |
| **query()** | $querystring | Perform query. Return object array of results or false if query returned an error. No result will return an empty array.  Query with no result (only true or false, such as INSERT) will return true if are successfully performed |

## User()

|  |  |  |
| --- | --- | --- |
| **Method** | **Parameters** | **Description** |
| **\_\_constructor()** | $username | Initialize user with username, password, email, user type, family tag.  If user does not exist or an error occur during query operation, $this->user = false; |
| **exists ()** | void | If fetching operations from db have been performed successfully, it returns true. Other ways it returns false. |
| **getUser()** | Void | Return $this->user |
| **controlValidation()** | $password | Check if $password is equal to  $this->user->password (case sensitive) return true if they are equal. |
| **createUser()** | $username, $password, $email, $family | Inserts into table users a new user. Returns false if an error occur. Else it returns true |

## Family()

|  |  |  |
| --- | --- | --- |
| **Method** | **Parameters** | **Description** |
| **\_\_constructor()** | $tag | Initialize user with username, password, email, user type, family tag.  If user does not exist or an error occur during query operation, $this->user = false; |
| **exists ()** | void | If fetching operations from db have been performed successfully, it returns true. Other ways it returns false. |
| **getThingList()** | void | Return array of Thing type object that contains all things connected to the family |
| **getUserList()** | Void | Return array of User type object that contains all users connected to the family |
|  |  |  |

## Thing()

|  |  |  |
| --- | --- | --- |
| **Method** | **Parameters** | **Description** |
| **\_\_constructor()** | $tag | Initialize user with username, password, email, user type, family tag.  If user does not exist or an error occur during query operation, $this->user = false; |
| **exists ()** | void | If fetching operations from db have been performed successfully, it returns true. Other ways it returns false. |
| **getThing()** | Void | Return $this->thing, which contains:  -tag  -family tag  -name  -user\_type (id)  -access\_level |

# Api

## Login

**Request method:** POST;

**Data:** data{user, pass};

### Error

* **User does not exist:** die and return error to index.html
* **Wrong password:** die and return error to index.html

### Success

* **Auth performed:** create session variable $\_SESSION[“user”];

## Signup

**Request method:** POST;

**Data:** data{user, pass, email, family};

### Error

* **User does not exist:** die and return error to index.html
* **Wrong password:** die and return error to index.html

### Success

* **Auth performed:** create session variable $\_SESSION[“user”];