Contents

[Files: 2](#_Toc436306596)

[InputRecheck.php 2](#_Toc436306597)

[Relog.php 3](#_Toc436306598)

[Login.php 4](#_Toc436306599)

[UserHandler.php 5](#_Toc436306600)

[Database.php 7](#_Toc436306601)

**API**

## Files:

1. Brand.php
2. Category.php
3. CheckoutSave.php
4. Database.php
5. Filter.php
6. ImageScaling.php
7. InputRecheck.php
8. Login.php
9. Mail.php
10. Orderdetails.php
11. OrderFilter.php
12. Orders.php
13. PDFGen.php
14. Product.php
15. Relog.php
16. SearchModel.php
17. Statistics.php
18. UserHandler.php
19. Viewcounter.php

### Brand.php

Author: Matus Kokoska

Handles brands table in the database.

#### Includes:

* Database.php

Class: Brand

#### Class params:

1. Private
   * handlerDB - instance of database handler
2. Public
   * Id - brand id
   * error - error message
   * name - name of brand

#### Functions:

1. Private
   * -
2. Public
   * construct(id=null) - creates a BrandHandler instance with optional id paramater
   * load() - if id is set, loads existing Brand
   * save() - saves edited properties of brand or creates a new brand if id is not set
   * delete() - deletes existing brand
   * getAllBrands() - returns an array of all brands in the database

### Category.php

Author: Matus Kokoska

Handles categories table in the database.

#### Includes:

* Database.php

Class: Category

#### Class params:

1. Private
   * handlerDB - instance of database handler
2. Public
   * id - id of category
   * error - error message
   * name - name of category
   * parent - super category of current category
   * alldescendants - an array of all children categories

#### Functions:

1. Private
   * getAllCategories() - returns an array of all categories in the database
2. Public
   * construct(id=null) - creates a CategoryHandler instance with optional id parameter
   * load() - if id set, loads existing category
   * save() - saves edited properties of category or creates a new category if id is not set
   * delete() - deletes existing category
   * getDescendatns(id) - creates an array of all children of current category
   * createSubs() - creates an array, where parent id is an index of of another array of all its children
   * getCatPath(id) - creates an array of category path from current category to its super parent
   * getCategories() - creates an array of all categories and its subcategories

### CheckoutSave.php

Author: Peter Varholak

Saves the order and cart content in the database. Sends an email to users email with supplement generated bill in pdf format.

#### Includes:

* Orders.php
* PDFGen.php
* Orderdetails.php
* Mail.php

Class: -

### Database.php

Author: MatusKacmar

Class: DBHandler

#### Class params:

1. Private
   * dbHost - database host name
   * dbName - database name
   * dbuser - user access name
   * dbPassword - password to access database
   * dbHandler - PDO handler
   * error - error message
   * statement - SQL query
2. Public
   * -

#### Functions:

1. Private
   * -
2. Public
   * construct() - creates an instance of DatabaseHandler
   * query(query) - prepares SQL statement for execution
   * bind(parameters,value,type=null) - binds the value to parameter
   * execute() - executes statement
   * resultSet() - returns result set rows from table
   * singleRecord() - returns single record from table
   * rowCount() - returns number of affected rows by previous action
   * lastInsertId() - returns the last inserted id as string
   * beginTransaction() - starts a transaction to run multiple changes in database at once
   * endTransaction() - ends transaction
   * cancelTransaction() - cancels transaction
   * dumpStatementParams() - get rid of binded params in statement

### Filter.php

Author: Matus Kokoska

#### Includes:

* Database.php
* Category.php

Class: Filter

#### Class params:

1. Private
   * handlerDB - instance of database handler
2. Public
   * availableCriteria
   * criteria
   * name
   * minprice
   * maxprice
   * category
   * brand
   * productArray

#### Functions:

1. Private
   * nameQuery()
   * sortQuery() - prepares part of query needed for sorting results. Sorts by “rating” on default
   * brandQuery() - prepares part of query responsible for brand filtering. By default selects all brands
   * selectedProductsQuery() - prepares part of query needed for sorting specified results in this->productArray[]
2. Public
   * construct()
   * categoryQuery() - prepares part of query responsible for category filtering. By default is ELECTRONICS
   * getResults() - get result from database by the prepared query
   * getBrands() - get available brands for filtering
   * getMaxPrice() - get maxprice of available products

### ImageScaling.php

Author: Matus Kacmar

#### Includes:

* -

Class: ImageScaling

#### Class params:

1. Private
   * productItemThumbnailWidth
   * productItemThumbnailHeight
   * productPreviewImageWidth
   * productPreviewImageHeight
2. Public
   * -

#### Functions:

1. Private
   * -
2. Public
   * productItemTumbnail(productid) - scales product item thumbnail
   * productItemTumbnailMargin(size) - calculates margin of product item thumbnail image
   * productPreviewImage(productid)
   * productPreviewImageIndex(productid,index)

### InputRecheck.php

Author: Tomas Paronai

#### Includes:

* Database.php

Class: Recheck

#### Class params:

1. Private
   * handlerDB- instance of database handler
2. Public
   * -

#### Functions:

1. Private
   * -
2. Public
   * construct() - creates an instance of Recheck
   * CheckEmail(email,size) - validates email
   * CheckInput(input,size) - validates input
   * dumpsSpecialChars(input) - clears the input from unwanted characters

### Login.php

Author: Tomas Paronai

#### Includes:

* Database.php

Class: Login

#### Class params:

1. Private
   * handlerDB - instance of database handler
   * name - the name of the logged in user
   * id - the id of the logged in user
   * username - logged in users name
2. Public
   * -

#### Functions:

1. Private
   * -
2. Public
   * construct() - creates an instance of Login
   * checkLogin(email,password) - compares the email and password if they match the data in database
   * getName() - returns logged in users name
   * getUserId() - returns logged in users id

### Mail.php

Author: Matus Kacmar

#### Includes:

* PHPMailerAutoload.php

Class: Mail

#### Class params:

1. Private
   * mail
   * host - mail sender host
   * user - sender email
   * password - password to access host
2. Public
   * -

#### Functions:

1. Private
   * -
2. Public
   * construct() - creates an instance of Mail
   * addRecipient(recipient) - add recipient addresses to mail
   * attachement(path,name) - add attachement to mail
   * composeMail(subject,body,alternativeBody) - compose email
   * sendMail() - sends the email and returns confirmation

### Orderdetails.php

Author: Matus Kokoska

#### Includes:

* Database.php

Class: Orderdetail

#### Class params:

1. Private
   * handlerDB - instance of database handler
2. Public
   * id - id of current detail
   * error - error message
   * productid - id of product
   * ordered - id of order
   * quantity - number of current detail
   * detailprice - price of the single current detail

#### Functions:

1. Private
   * -
2. Public
   * construct(id=null) - creates an OrderdetailHandler instance with optional id
   * load() - if id is set, loads existing orderdetail
   * sace() - saves edited properties of orderdetail or creates a new orderdetail if id is not set
   * delete() - deletes existing orderdetail

### OrderFilter.php

Author: Matus Kokoska

#### Includes:

* Database.php

Class: OrderFilter

#### Class params:

1. Private
   * handlerDB - instance of database handler
   * querycounter
2. Public
   * userid
   * name
   * surname
   * shipped

#### Functions:

1. Private
   * -
2. Public
   * useridQuery() - prepares a part of query for filtering orders by userid
   * nameQuery() - prepares a part of query for filtering orders by name
   * shippedQuery() - prepares a part of query for filtering orders whether they are shipped or not
   * surnameQuery() - prepares a part of query for filtering orders by surname
   * getResults() - get results from database as an array of orderids

### Orders.php

Author: Matus Kokoska

#### Includes:

* Database.php

Class: Brand

#### Class params:

1. Private
2. Public

#### Functions:

1. Private
2. Public

### PDFGen.php

Author: Matus Kokoska

#### Includes:

* Database.php

Class: Brand

#### Class params:

1. Private
2. Public

#### Functions:

1. Private
2. Public

### Product.php

Author: Matus Kokoska

#### Includes:

* Database.php

Class: Brand

#### Class params:

1. Private
2. Public

#### Functions:

1. Private
2. Public

### Relog.php

Author: Tomas Paronai

Relog handles registration, login and user profile edit.

#### Includes:

* Database.php
* InputRecheck.php
* UserHandler.php
* Login.php

Class: -

#### Functions:

1. Private
   * -
2. Public
   * errorControl(name,surname,email,password)

**errorControl(name,surname,email,password)**

Validates user inputs for registration form.

**@author** Tomas Paronai

**@param** $name

**@param** $surname

**@param** $email

**@param** $password

**@return** - true if all OK

### SearchModel.php

Author: Matus Kokoska

#### Includes:

* Database.php

Class: Brand

#### Class params:

1. Private
2. Public

#### Functions:

1. Private
2. Public

### Statistics.php

Author: Matus Kokoska

#### Includes:

* Database.php

Class: Brand

#### Class params:

1. Private
2. Public

#### Functions:

1. Private
2. Public

### UserHandler.php

Author: Tomas Paronai

#### Includes:

* Database.php

Class: User

#### Class params:

1. Private
   * handlerDB - instance of database handler
   * success – data save status indicator
   * id – the id of the current user being handled
2. Public
   * -

#### Functions:

1. Private
   * -
2. Public
   * editUser(id)
   * newUser(name,surname,email,password)
   * saveFirstData(parameter,value)
   * saveData(parameter,value)
   * setId(id)
   * getId()
   * getValidId()
   * getData(parameter)
   * isSaved()
   * delete()

**editUser(id)**

Creates an UserHandler instance with only the given ID

**@author** Tomas Paronai

**@param** $id

**newUser(name,surname,email,password)**

Creates an UserHandler instance with only the given params

**@author** Tomas Paronai

**@param** $name

**@param** $surname

**@param** $email

**@param** $password

**saveFirstData(parameter,value)**

Saves the first data in the table.

**@author** Tomas Paronai

**@param** $parameter - name of the colum

**@param** $value

**saveData(parameter,value)**

Edits user info by the given id | saves the user input in the database.

**@author** Tomas Paronai

**@param** $parameter - name of the colum

**@param** $value

**@param** $id - given id

**setId(id)**

**@author** Tomas Paronai

**@param** $id

**getId()**

**@author** Tomas Paronai

**@return** $id

**getValidId()**

**@author** Tomas Paronai

**@return** - id of the user which is being handled

**getData(parameter)**

Returns the data from database of the current which is being handled.

**@author** Tomas Paronai

**@param** $parameter - the data i want

**@return** $users[$i][$parameter] - data at parameter

**isSaved()**

**@author** Tomas Paronai

**@return** "User registered." - user was successfuly saved in the database

**@return** "Registration failed." - saving failed. Probably database error.

**delete()**

On call deletes the current user from database.

**@author** Tomas Paronai

### Viewcounter.php

Author: Matus Kokoska

#### Includes:

* Database.php

Class: Brand

#### Class params:

1. Private
2. Public

#### Functions:

1. Private
2. Public