Contents

[Files: 2](#_Toc444846874)

[Brand.php 3](#_Toc444846875)

[Category.php 3](#_Toc444846876)

[CheckoutSave.php 4](#_Toc444846877)

[Database.php 4](#_Toc444846878)

[Filter.php 5](#_Toc444846879)

[ImageScaling.php 6](#_Toc444846880)

[InputRecheck.php 7](#_Toc444846881)

[Login.php 7](#_Toc444846882)

[Mail.php 8](#_Toc444846883)

[Orderdetails.php 9](#_Toc444846884)

[OrderFilter.php 9](#_Toc444846885)

[Orders.php 10](#_Toc444846886)

[PDFGen.php 11](#_Toc444846887)

[Product.php 12](#_Toc444846888)

[Relog.php 13](#_Toc444846889)

[SearchModel.php 13](#_Toc444846890)

[Statistics.php 14](#_Toc444846891)

[UserHandler.php 14](#_Toc444846892)

[Viewcounter.php 15](#_Toc444846893)

****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

Handles connection with the server to database. Uses PDO driver.

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

Handles filtering products and orders.

#### 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

Scales the images to the required size for product preview and thumbnail.

#### 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

Validates user inputs.

#### 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

Handles user login.

#### 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

Handles email sending and email generation.

#### 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

Handles orderdetails table in the database.

#### 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

Handles orders filtering.

#### 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

Handles orders table in the database.

#### Includes:

* Database.php

Class: Order

#### Class params:

1. Private
   * handlerDB - instance of database handler
2. Public
   * Id - order id
   * error - error message
   * userid - id of user
   * surname
   * name
   * email
   * phone
   * address
   * city
   * postcode
   * datecreated - date when the order was created
   * shipped - 1 shipped, 0 queued
   * orderprice
   * alldetails - array containing Orderdetails

#### Functions:

1. Private
   * getDetails() - get all Orderdetials for the current order
2. Public
   * construct(id=null) - creates an OrderHandler instance with optional id
   * load() - if id is set, loads existing order
   * save() - saves edited properties of an order or creates a new order if id is not set
   * delete() - deletes existing order

### PDFGen.php

Author: Tomas Paronai

Handles pdf file generation.

#### Includes:

* mpdf.php
* Orders.php
* Orderdetails.php
* Product.php

Class: pdfFile

#### Class params:

1. Private
   * order - order object currently handling
   * pdfFile - mpdf object
   * htmlOut - body of the pdf file in html5
   * pdfId - file name, generated from ordered, userid and date when the order was created
   * absolutePath - path to file (libraries/pdf/pdfId.pdf)
2. Public

#### Functions:

1. Private
   * createStyle() - appends stylesheet to pdf file
   * createHead() - appends header to htmlOut
   * createAddress() - appends addresses to htmlOut
   * createSum() - appends all details and price to htmlOut
   * createFoot() - appends footer to htmlOut
   * finish() - writes the htmlOut to pdfFile, generates pdf and saves it to the given absolute path
2. Public
   * buildPDF() - call steps to generate pdf file
   * getPath() - returns the path to pdf file

### Product.php

Author: Matus Kokoska

Handles products table in the database.

#### Includes:

* Database.php

Class: Product

#### Class params:

1. Private
   * handlerDB - instance of database handler
2. Public
   * id - product id
   * error - error message
   * categoryid - id of category where the product belongs
   * amount - quantity in warehouse
   * name
   * price
   * brandid - id of brand where the product belongs
   * description
   * viewamount - number of views of current product
   * datecreated
   * numofratings - number of ratings on current product
   * sumofratings - sum of ratings on current product
   * saved - if true, file was successfully saved/edited
   * deleted - if true, its archived

#### Functions:

1. Private
   * -
2. Public
   * construct(id=null) - creates a ProductHandler instance with optional id
   * load() - if id is set, loads existing product
   * save() - saves edited properties of product or creates a new product if id is not set
   * delete() - archives existing product
   * getTotalProducts() - get total number of products in database
   * getTotalArchived() - get total number of archived products in database
   * getProductsId() - get an array of all product ids in database

### Relog.php

Author: Tomas Paronai

Relog handles registration and login. If registration was successful, it generates and sends an activation email to the new user.

#### Includes:

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

Class: -

#### Functions:

1. Private
   * -
2. Public
   * errorControl(name,surname,email,password) - validates user inputs for registration form

### SearchModel.php

Author: Matus Kokoska

Handles search in database based on user input table in the database.

#### Includes:

* Matus Kacmar

Class: SearchModel

#### Class params:

1. Private
   * searchTerm - user input
   * handlerDB - instance of database handler
   * prepare - input validator
2. Public
   * -

#### Functions:

1. Private
   * -
2. Public
   * getResults() - get results related to search term from database
   * foundResults(results) - find out if query returned some results
   * getSpecificResults(table, term) - creates a query based on term array for table param

### Statistics.php

Author: Matus Kokoska

#### Includes:

* Database.php
* Product.php

Class: Statistics

#### Class params:

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

#### Functions:

1. Private
2. Public
   * getProductCount() - get total number of not archived products
   * getUserCount() - get total number of activated and not archived users
   * getShippedOrderCount() - get total number of shipped orders
   * getNotShippedCount() - get total number of not shipped orders
   * getBestSellingProduct() - get the id of the most bought product

### 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
   * randomPass - generated password
2. Public
   * -

#### Functions:

1. Private
   * generatePassword() - generates a random password [0-9a-zA-Z]
2. Public
   * editUser(id) - creates an instance of existing user
   * newUser(name,surname,email,password) - creates an instance of new user
   * saveFirstData(parameter,value) - saves first data of new user to users table to given parameter
   * saveData(parameter,value) - saves data to users table to given parameter
   * setId(id) - sets the id
   * getId() - returns current id
   * getValidId() - returns next valid id
   * getData(parameter) - returns data from users table of given parameter
   * isSaved() - returns true if data saved
   * delete() - archives current existing user
   * restore() - restores current existing user
   * getGenPassword() - returns generated password

### Viewcounter.php

Author: Matus Kokoska

#### Includes:

* Database.php

Class: Viewcounter

#### Class params:

1. Private
   * file
   * fileContent
2. Public
   * -

#### Functions:

1. Private
   * read() - reads data from existing text file viewcount.txt
   * write() - writes data to text file viewcount.txt
2. Public
   * increment() - increments the number of page view on a current date
   * getTotalViews() - counts total view amount
   * chartDataProvider() - provides data for javascript chart of page views