



miro



Filesystem Explorer

PHP Local FileSystem Explorer

Juan Carlos Cabello
Javier Fernández

1. General analysis



Local Package manager



Frame 1 IMPORTANT THINGS

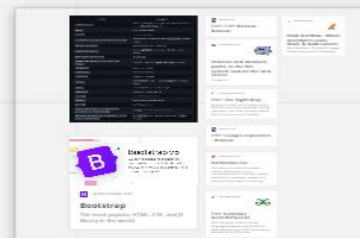


Diagram functions create, modify and delete directories

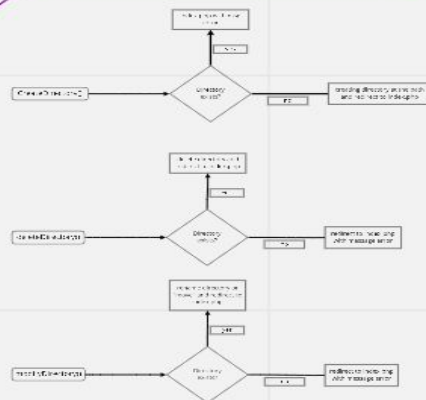
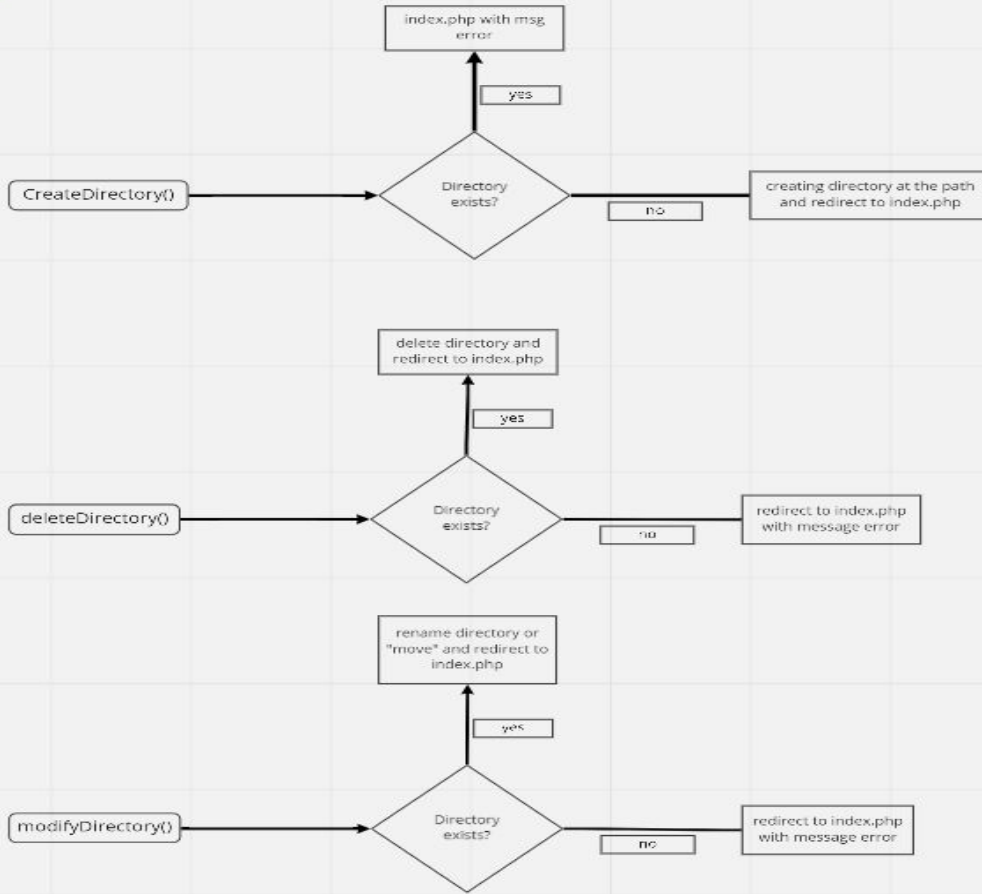


Diagram UML



We use Miró Application and UML Diagrams to organize and develop the code behaviour, to make easier the develop phase of the project.

<https://cacoo.com/es/>

<https://app.diagrams.net/>

<https://app.moqups.com/sign-up>

1. General analysis

Requirements

You cannot use file in a party browser.
You will not be able to add a new file to a folder.
You must use 2D.
You must use the file in a 2D.
Create a new file in a party browser.
Get the data from the server to show the folder in a browser.
Use the same code to create a new folder in a browser.
In the case of using different programming languages, always define the
organization - requirements.
Remember that it is important to divide the work into several sub-tasks that it
is easy to understand each participant role of the team with a specific
code.
You should try to make as possible for the team and the client and use the
same code that they are not used or are not necessary to evaluate the project.

Step 1: Analyze the project requirements

- Create, modify and delete directories
- Browse through directories from an initial party
- The first path will start from a folder inside the project repository and the file will be "root"
- Specify a directory and file by name
 - In the case of new string files by type, you must also be able to specify the expected output of the name.
- Navigate through the initial path established and all the folders created from that path. The folder must not be able to use or navigate to the parent folder of the file to be.
- List all files in a directory
- See the following information of files and directories:
 - Creation date
 - Last modified date
 - Size of the file
 - Size
 - If it is less than 1 MB show KB, otherwise show MB
- Show the last of the main file extensions such as:
 - doc
 - xls
 - ppt
 - txt
 - json
 - pdf
 - zip
 - rar
 - exe
 - img
 - mp3
 - mp4
- Show the associated images
- Play the associated videos
- Play the associated audios

Step 2: Design the project structure

- How will the interface be
- You will have to design a scheme of your application taking into account the requirements
- What actions can be executed by the user
 - You will have to design a new user diagram
- Analyze and understand what brings more value to the user
- Analyze how you will organize the project at the level of directories and files.

Step 3: Start to develop the project

Once you have all the designs and organization of your project (needs), you must start developing it.

Step 4: Extra functionalities

- In addition to the requirements mentioned above, you can add the following extra functionalities so that the user is able to:
- Show the information of the selected "file" (file on the screen).
 - Show files and directories between folders.
 - When deleting a file or folder, it will be moved to a specific folder called "trash", you must be able to completely copy/delete it from the mentioned folder.

General analysis

Step 1: Analyze the project requirements

Step 2: Design the project structure

Step 3: Start to develop the project

Step 4: Extra functionalities

method	description
<code>clearDirectory()</code>	returns DirectoryIterator Object to list directory content as new <code>PHPFile</code> objects
<code>mkdir()</code>	try to create directory
<code>rmdir()</code>	try to remove directory
<code>recursiveDirectoryIterator::construct()</code>	wrap directory recursively to new <code>StackIterator</code> Storage adapter
<code>getFileInfo()</code>	get file modified into timestamp
<code>getFileInfo()</code>	same as <code>getFileInfo()</code> , but returns a file info object instead
<code>getFileInfo()</code>	calculate size
<code>getFileInfo()</code>	calculates a hash (md5) with <code>md5File()</code> algorithm
<code>isReadable()</code>	is directory readable?
<code>isExecutable()</code>	is directory executable?
<code>isWritable()</code>	is directory writable?
<code>isPath()</code>	is selected path a file? => always <code>FALSE</code> for Directory instance
<code>isDir()</code>	is selected path an actual directory?
<code>isFile()</code>	is directory a hidden directory?
<code>isLink()</code>	run recursive validation
<code>isLink()</code>	access internal storage adapter
<code>getPath()</code>	fetch filesystem path
<code>getFileInfo()</code>	get <code>FileInfo</code> in current directory by name
<code>getFileInfo()</code>	get <code>FileInfo</code> in current directory by name (could be storage)



Bootstrap v5

Quickly design and customize responsive mobile-first sites with the world's most popular front-end open source toolkit.

 getbootstrap.com

Bootstrap

The most popular HTML, CSS, and JS library in the world.


 www.php.net

PHP: PHP Manual - Manual

 phpdelusions.net

Relative and absolute paths, in the file system and on the web server.

Intro#intro


 phptherightway.com

PHP: The Right Way

An easy-to-read, quick reference for PHP best practices, accepted coding standards, and links to authoritative PHP tutorials around the Web

 www.php.net

PHP: Escape sequences - Manual

 necolas.github.io

Normalize.css

Normalize.css makes browsers render all elements more consistently and in line with modern standards. It precisely targets only the styles that need normalizing.

 www.geeksforgeeks.org

PHP Tutorials - GeeksForGeeks

PHP is a server-side scripting language designed specifically for web development. PHP can be easily embedded in HTML files.

 stackoverflow.com

Stack Overflow - Where Developers Learn, Share, & Build Careers

Stack Overflow | The World's Largest Online Community for Developers

Documentation

<https://www.php.net/manual/en/>

<https://phpdelusions.net/articles/paths#commands>

<https://phptherightway.com/>

<https://www.php.net/manual/en/regexp.reference.escape.php>

<https://necolas.github.io/normalize.css/>

<https://www.geeksforgeeks.org/php-tutorials/?ref=lbp>

<https://getbootstrap.com/>

<https://stackoverflow.com/>

<https://github.com/assembler-institute/php-manage-files-workshop/tree/master>

What lessons you've learned during this project

- *Improving your problem-solving skills*
- *Improving your teamwork and communication skills*
- *Increasing your knowledge of programming concepts and languages*
- *Creating awesome projects to showcase your work*
- *Focusing on writing clean efficient code*

What problems have you encountered when developing this project?

- **Time** => One of the main problems of any project is the distribution of work/time, in this case we have distributed it in an objective way when carrying out the approach in "Miro", to avoid surprises.
- **Project scope** => The general approach of the project has been to divide it into 3 main phases:
 - 1) Overview and organization.
 - 2) Distribution, documentation, and content.
 - 3) Execution, review, and cleaning of code.
- **Budget** => Thanks to [Assembler Institute](#), our budget has not been a problem, we have unlimited funds to carry out our project.

How you have organized and distributed the tasks

1. *Create, modify and delete folders.*
2. *Browse directories from initial path.*
3. *Navigate through the initial path to the folders and existing files.*
4. *Upload a file to a directory.*
5. *Search Directories and files by name.*
6. *See the creation date, last modified, extension and size.*
7. *Show the icon depending the extension file or folder.*
8. *View images, play videos and audios.*
9. *Move files to trash and delete from it.*
10. *Make a chart of sizes for each type of fyle.*
11. *Luckily for us, we have distributed the main functions in two phases: 1-A colleague created the code and the other looked for everything necessary to continue effectively without stopping until we achieved our goals.Efficiency, Speed, Peer-Coding and meeting objectives.*