Linnaeus University

1DV608 - Web development with PHP
Project FileUpload

Student: Benjamin Björk
Student ID: bb222bx@student.lnu.se

Project vision
My vision is to make a webpage where a user can upload a file for any reason. I wanted to make a site where I could dump temporary files without using a ftp or sftp client or if the computer is not allowed to use any file transfer programs.

Requirements of the project
This is the requirement that must be on the FileUpload site.

Requirements
- *Public file upload:* User can upload files without logging in to the site.
- *Public uploaded files* must be accessible for not logged in user.
- *Public file list:* Not logged in users gets a public file list on the main page.
- *Private upload function:* User must be able to upload files that becomes his/her and no one else can access the files.
- *Login function:* User must be authenticated for being able to upload private files.
- *Private file list function:* Logged in user must see his/her uploaded files.
- *Public/Private users* must be able to download files from the server.
- *Security:* Users must not be able to access the upload folder if not authenticated.
- *Database* is required.
- *No javascript.*
- Code has one single page.
- No external libraries is allowed.
- MVC implementation.
- Code must be available on public git repository.
- The application must be deployed on a public server.

Implementation of the requirements
I started with the first requirement "*Public file upload*". It was hard to decide if I should upload the file to the database as a BLOB data or as a file to a folder in the server. I decided that it would give me better performance to save the file to the server in a upload folder. This is because I run my web server on a *raspberry pi*. It should be more secure to upload everything to the database as a BLOB but I had already implemented the save to folder function so I didn't change the upload function. To future update on the application all private upload should be uploaded as a BLOB data to the database.
This is how the function works:
User chose file and click upload. The page responds with successful message if uploaded correctly and shows file information to the user. The page shows a link to the file if user want to see the file directly.
All files is saved to a public upload folder and the path, file type, size is saved to the database.

Public file list
When a user enter the page the server loads data from the database such as file type, file path, size and shows a list of the files to the user. The filename is a link to the file on the server.
If users click on the filename the server loads the file and the web browser shows/download the file.

Sign up
To be able to upload private file I must have a registration function for the user to register a new account on the site.
I used MySQL database for saving user information and to retrieve user credentials.
I added basic error checking on user input such as: Username/password can't be empty, password don't match (password field and a retype password field), username is taken.
Create user if all checks is okey.

Sign in
Users must sign in for being able to upload a private file.
In this function I added basic error checking such as: check if username/password is empty, username/password is wrong.

If a user writes wrong credentials the "wrong username and password" message will be shown in the sign in page.

Private file list
When the user login the application will show the user's private file uploads in a "Private file list" on his/her member area page.
This is done by using the database. All private files is uploaded with these data:

- user id
- username
- file type
- size

By checking the logged in user from the session I can search for files that is uploaded by that user and present a file list to him/her.

Private file uploads
Logged in users will always do private uploads, this is to prevent upload publicly by mistake.
This can be changed in future update of the application.

Security
The application have basic SQL injection protection by using PDO and prepared statements.
The server have disabled directory listing so a user can't list files on the upload folders.
Public and Private uploads are in different folders. The folders have read and writes attributes so the files can be downloaded by a user.
The database configuration file is located in /var/conf/ so the password for the database isn't accessible from internet.

External library and javascript
No external libraries are used neither any javascript.

Implementation

I implemented one function at a time and tested it so it worked, then made the next function that is connected to the previous function.
Example: I made the view of the start page (show public list) then made the public upload function. Tested with dummy files and then made the public file list function that lists the files uploaded to the server.
I didn't brainstorm so much about the project. All I first wanted was to upload files to a server, as time goes by and the upload function is finished I wanted a "Private upload" function so I had to implement a login system so user i authorized and can upload private files.

I used MVC on this application, all outputs and $_GET, $_POST is done in view classes and is sent to model when needed.
I have two controllers in this application. The first controller is Master Controller, it controls if user want to upload files or when user want to login.
The second controller is for login , it controls the login system.
I have different models for different function, there are:

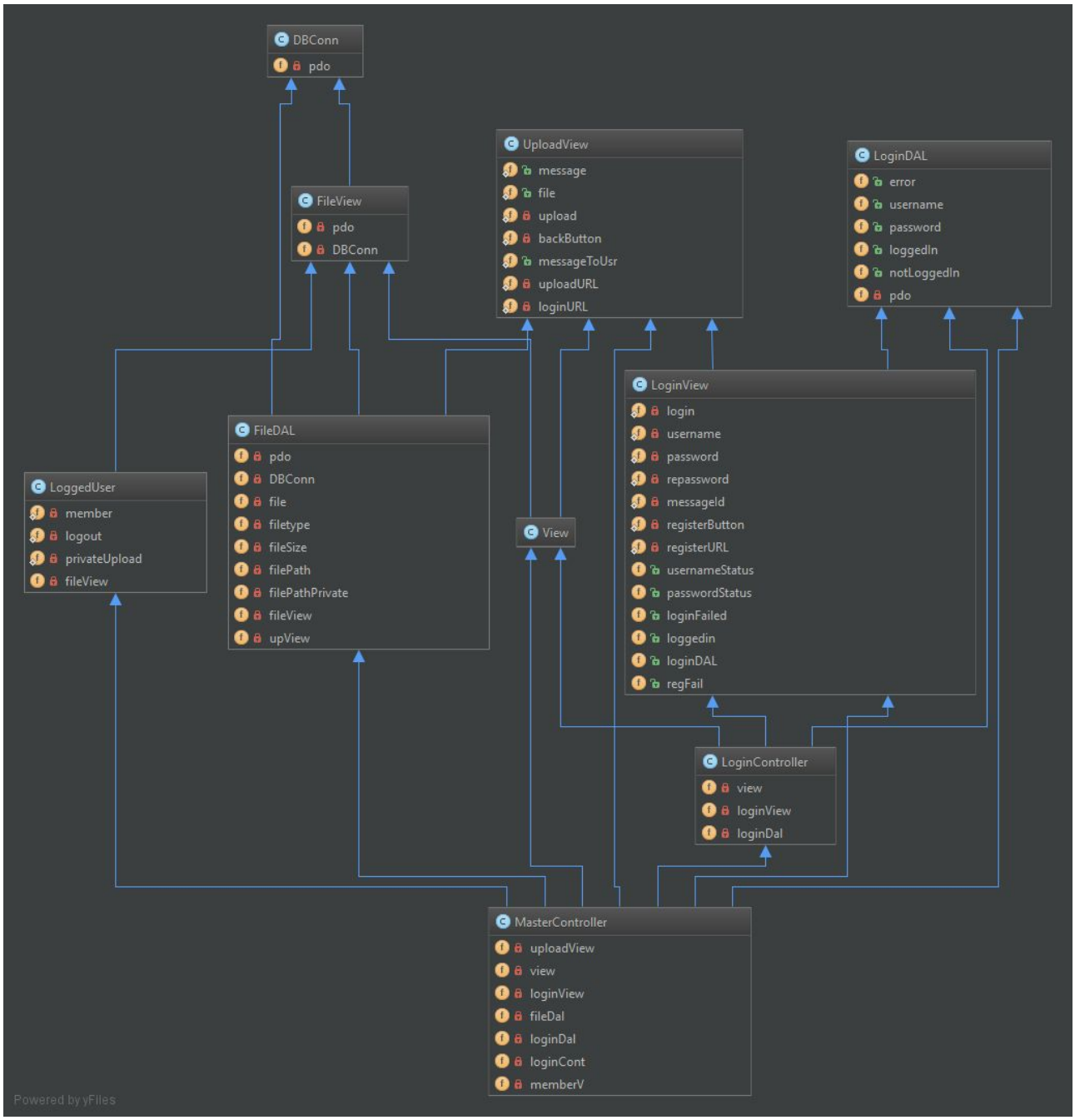
- Login model
- File upload model
- Database model

I didn't focus on the use-case directly because I implemented more functions when I wanted more from the application. So the use-cases was made after the function was implemented and tested. When a function was finished I made use-case for that function about how it should work and think about what the user want from this function.

Test-process

I tested the application every time I made a change, so tests was made immediately, because I wanted the application to work as it should and I wanted to see so it did. This is how I could ensure good code quality. I have commented the code so it is easier for a new person to look at the code and understand what the different part do and how they are connected to each other.

UML Diagram of the code



Discussion

When I started this course I didn't have any PHP knowledge but after this course I have much better understanding on how MVC works and why it is good way to code.

With MVC I can make modules to an application without affecting the whole application. If I wanted to change a module of the application I just have to edit the module name in the index and edit the dependencies if it have dependencies.

The security of the application

The security isn't the best and this application should not be used on a production environment as is.

I use ".htaccess" to prevent directory listing on the server and is easy to bypass if you are an attacker.
The login credentials is not encrypted! This is because I didn't have time to implement an encryption function. I have looked at "bcrypt" and the next thing to implement would be use bcrypt in the application.

Future work
I want to use CSS and give the application a nice design and use javascript to make the site more responsive.
Functions that I want to implement:

- Encrypt user credentials and private files
- Uploading progress bar
- Share links by mail, to facebook
- Search for files
- Copy files from remote servers to this server. Example, copy a picture from "Google picture" to a private file list.
- Remove files from public file list automatically after a period of inactivity
- Remove files from private file lists.