Chapter I
Forewords

History of communication is as old as the history of Humanity and Mankind has managed to evolve it along centuries throughout incredible revolutions.

In 1794, Claude Chappe tried to resolve the issue of long-distance communication,that was limited to horse speed at that time. He set up an ingenious communication system of air telegraph during the French Revolution. Chappe's "tours" (towers) were covered by a mobile mast that could be seen with binoculars from the nearest neighbor tower located at approximately 10 to 15 km.

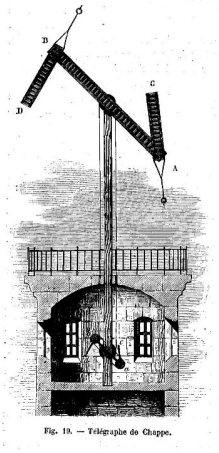


Fig. 19. — Télégraphe de Chappe.

The Paris-Lille line was operational the same year and allows transmissions between the two cities at a speed of 9 minutes per letter throughout 15 towers. Obviously, trans-mission time depends on its length.

In 1844, 534 towers were dispersed on the French territory linking on more than 5000km, the most important cities.

But, this system had 2 big disadvantages: it couldn't be functional by night because,obviously, of the bad visibility, and the number of operators per tower (2 every 15 km).

Fortunately, we are in the 21st century.

Chapter II
Introduction

Now you are ready to build your first web applications, like pros. If you didn’t mind, the web is a vast and rich world, allowing you to release data and content quickly to everyone around the world. Now you know the basics, here comes the time to leave those old fashioned to-do lists and eBusiness websites, and fly away toward bigger projects. You will discover new notions and the beauty of: Responsive design，DOM Manipulation，SQL Debugging，Cross Site Request Forgery, Cross Origin Resource Sharing.

Camagru's Specifications:

Made by Henricus Wouter de Vos (Slack: @hde-vos)

Note:

*Remember to read everything, not just the requirements. All data in this document has been written so that it is relevant and accurate to the PDF's requirements.
*Guest: not logged in user
*All abbreviations will have a single instance where their full wording will be shown. After that, it's up to you to remember it.

Overview: Create a small web application to make basic photo and video editing using your webcam and some predefined images. Users should be able to select an image in a list of superposable images, take a picture with their webcam and admire the result that should be mixing both pictures. All pictures should be public, likeable and commentable.

General instructions:

- Submit a file called "author" that contains creators username at the root of your repo.
¯_(⊙_⊙)_/¯
- Application must produce no errors, warning or log line in any console, server or client side. getUserMedia related errors are tolerated. ʘ_ʘ *??? Obviously???*
- Must use PHP with just the standard library.
- Pages must use HTML, CSS and Javascript.
- Only Frameworks that you created are allowed, except for CSS frameworks that doesn't need **forbidden** JavaScript.
- Must use PDO ("PHP Data Objects ") abstraction driver to communicate with your database ("DB") and DB must be queryable. Error mode of this driver must be set to PDO::ERRMODE_EXCEPTION.
See: (What is PDO), (How to PDO???), (How to PDO??? But better than previously)
- Application must not have any leaks, whether security or memory leaks. See: Security Leaks
- Any web server can be used. Apache, Nginx or PHP's Built-In Server. See: PHP's Built-In Server
- App must be compatible with at least Firefox (>=41) and Chrome (>=46).

Common features:

- Develop a MVC structured web application.

MVC??????

M stands for **Model** and relates to the background ie. your Database.

V stands for **Visual** and relates to the frontend of your website ie. HTML.

C stands for **Controller** and relates to the "code" that access the **Model** and sends it to the **Visual** ie. controller sends data from the database to be displayed on the WS.

- The website should have a decent layout (Guidelines that the pdf provided is having a visual Header, Body and Footer for your website. See: almost any modern website layout.), function normally on a smaller resolutions and phones.

- All forms should have validations and security. **THIS IS MANDATORY!!!! THIS WILL BE TESTED DURING EVALS.**

Things that are considered unsecure:

*Admins are able to see database's passwords due to being stored in plaintext.
*User is able to inject SQL-code
*User has the ability to upload unwanted content onto server.
*User is able to alter a SQL-Query
*User is able to use an extern form to manipulate private data.

See: SQL Injection, More SQL Injection, Even More SQL Injection

Solutions:

*Encrypt password in database.
*PHP has inbuilt functions that prevents SQL-injection attacks.
**********Gabby, were fourth are though?????
*PHP has inbuilt functions that prevents SQL-injection attacks.
*Not 100% sure on how to prevent this yet. More research to be done.

User Features:

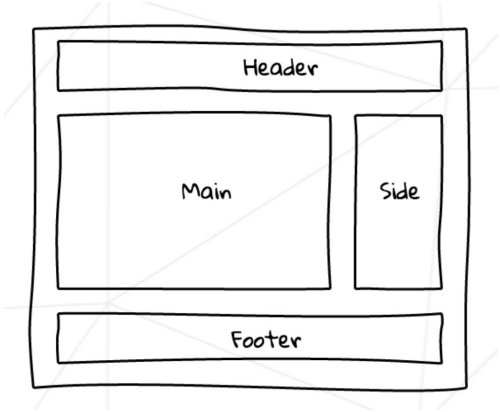
- User should be able to sign up with minimum levels of complexity. They must have a **VALID** email, username and password (According to ~@agabrie, the marking sheet requires the password to be complex).
- User have to validate their account through a validation link sent to their email.
- User should then be able to login to your application through their username and password. User should also be able to request a password change. The user should receive an email with a link to a webpage where they can change their password.
- User should be able to logout anytime and in one click on any page.
- *This next point in the pdf is very weirdly worded, so I am assuming a bit here and taking a bit of a guesstimate. If you can give an alternative meaning, please message me (@hde-vos). It reads: "Once connected, an user should modify his username, mail address or password."*
    * Once a user logs in, they should be able to modify their username, email and password. (ಠ ಠ)

Gallery Features:

- The Gallery has to contain ALL images posted by anyone, ordered by date of creation. Only a logged in user should be able to like and comment on the pictures. PDF does not specify whether the Gallery should start at creation date or end at creation date. It makes more sense that the latest images are displayed here, otherwise the user will see the same 5 images every time they visit the Gallery.
- When an image receives a new comment, the OP of the image should be notified via email. Notifications about comments must be a preference set to enabled by default (Annoying features being set to enabled by default???? 42 is Facebook confirmed), but should be able to disable it in the users preferences.
- Gallery should have at least 5 images on a page meaning that if you have 20 images in your database, then you should have 4 pages and each page should have 5 images on them.

Editing features:

*Figure V.1: Layout idea from the PDF



Only logged in users should be able to access this page and any guest should be “gently” rejected.

This page should contain 2 sections:

- A main section containing the preview of the user’s webcam, list of superposable images and a button allowing to capture a picture.
- A side sections displaying thumbnails of all previous pictures taken.

Your page layout should look similar to Figure V.1.

- Superposable images must be selectable and if no superposable image has been selected then the button that captures the image should be inactive.
- Creation of the image must be done on the server side.
- User should also be able to upload an image that they want to edit through the editor.
- Users should be able to delete their own uploaded images.

Constraints and mandatory things:

Your application should respect the following technologic choices:

-Authorized languages

*[Server Side] PHP

*[Client] HTML - CSS - Javascript (Ont with browser natives API)

-Authorized frameworks

*[Server] None **(┘ ʘ_ʘ)┘ ≋┴──┴**

*[Client] CSS Framework tolerated, unless it adds forbidden JavaScript.

You project should contain imperatively:

- An **index.php** file at the root of your repo and being the entry point to your website.
- A **config/setup.php** file, capable of creating or recreating the database schema by using the info contained in config/database.php
- A **config/database.php file**, containing the database's configuration and will be instanced vir PDO in the following format:

```php
<?php
    $DB_DSN = ...;
    $DB_USER = ...;
    $DB_PASSWORD = ...;
```

DSN ("Data Source Name") contains required information needed to connect to the database, for instance 'mysql:dbname=testdb;host=127.0.0.1'. Generally, a DSN is composed of the PDO driver name, followed by a specific syntax for that driver. For more details take a look at the PDO doc of each driver. See: Documentation of the PDO constructor

Bonus Part:

You can start working of bonus' once all of the mandatory part has been completed. You should still respect the constraints imposed on the previous page. ಥ∩ಥ

- "AJAXify" exchanges with the server.
- Propose a live preview of the edited result, directly on the webcam preview. PDF says this is supposedly a lot easier than it may seem.
- Do an infinite pagination of the gallery part.
- Offer the possibility for a user to share their images on social networks, ie. share the image created on your website to other social media like Facebook, Reddit, Twitter etc.
- Render an animated GIF.

Other bonus ideas:

- Allow user to upload a Profile Picture
- Search for other users to view their profiles

Assessment and Peer-review:

Submit your work on your GIT repo as usual. Only the work on your repository will be graded.

The following requirements are evaluated in the scale. Be very attentive and consider them carefully as they will be sanctioned by a 0 if not respected.

Good luck and don’t forget to push! (つ ◕_◕)つ