# Simple login web page with administration

1. **Page descriptions**
2. Log-in page: simple login page, nothing too fancy. Has two input fields for username and password and a “Log in” button. The password field must have the ‘password’ input type.
3. User page: after successfull login, the user is navigated to a page with only menus on the top. The menus have a „hover” functionality to open sub-menus. Clicking on a bottom menu will change the page to have the menu name as a content (server-side). Users have a “log-out” button. Menus should be hyperlinks, but other versions are acceptable too.
4. Admin page: The admin page has similar menu to the user. the menu items in an admin page are:

c.1) Users management: The content of this page is a “Create new user” button, a list of all the users with an “Edit” and “Delete” button next to them.

- Create a new user menu entry: clicking it changes the content to a page for creating new users. It has a field for the name, two fields for the password and repeat password, a way to select the groups of the user(this has many variations: autocomplete, multiselect list, checkboxes, etc.), a checkbox for admin rights and a Create and Cancel buttons.

- Edit user: has the same fields as create new user, but the Create button is called Update.

c.2) Menu management: The content of this page is a “Create new menu” button, a list of all the menus with a “Edit” and “Delete” button next to them.

- Create new menu: has a field for the menu name, a drop-down list for its group (in which the user can type his own new group or choose one of the existing from the dropdown), a dropdown list for the parent menu (it contains all the menus and a blank entry at the top for no parent), a Create and Cancel buttons. **Each menu can have only one group.**

- Edit menu: has the same fields as create new menu, but the Create button is called Update.

- The Delete button has a confirmation action attached to it.

c.3) Groups management: The content of this page is a list of all the menus with a “Edit” and “Delete” button next to them. The Edit function will only change the group’s name.

**2. Database.**

The database will have all the users, menus and groups.

The user entries have a: user\_id, name, password (encrypted or not), is\_admin (bool field)

The menus have a: menu\_id, name, parent\_menu\_id, group\_id, action (for the click event).

The groups have a: group\_id, name

Because a user can have many groups, and a group can have many users, a junction table is needed. It will be called UserGroups and have two fields: user\_id and group\_id

**3. Server-side functionality.**

To be effective the server functionality will be split in functionally independent areas:

* Login and logout
* User creation, update and delete
* Menu creation, update and delete
* Group creation, update and delete
* User page
* Admin page

Others can emerge in the process of development.

An important part is the login functionality, which must be separate, clean and easy to enhance. A **policy of “create the basic functionality with the mindset to refactor it for improvement later”** should come in handy.

That is why **deciding on the most essential and basic needs for the requirements to be met should be a high priority**. No unneeded time should be spend on unnecessary things like CSS, session cookies, hover effects, performance, etc. This does **not** include trivial things like code and design quality.

Any access to or from the database are created through the model functionalities (by MVC standard):

* DB read
* DB modify