**Student : ZEROUAL AYMENE**

**Planning :**

Proposition 15 was chosen to be theme of the website, First Website's functionality and Data models was detailed and documented like this :

• Self-Installation in the first use :

- Database creation + Tables

- Creation of an Admin account with CEO privileges

• Admin management:

- Admin Model:

-ID: Unique identifier number

-Username: Admins username, used in authentication

-Password: Admins password, used in authentication

- IS CEO : Yes : Admin with CEO privileges

- **Only an admin with CEO privileges** can Add/Edit/Delete admins accounts

• Materials Management:

- Material Model:

-ID

-Name: Name of material

-IS Available(is\_free): Yes : material is available and can be rented

-Number of rents

-Clients list: Hold client’s id who rented this material

-Image path: Hold image file name , default image : default.png

- **Only logged in admin** can Add/Edit/Delete materials

• Clients Management:

- Client Model:

-ID

-First name

-Last name

-Email

-Phone number

-Status : Hold an int of violations + 1 , used to punish clients who reach specific number of violations

- **Only logged in admin** can Add/Edit/Delete clients

• Rents Management:

- Rents Model:

-ID

-Client ID : Holds client’s id who rented

-Material ID : Hold material’s id who got rented

-Price

-Creation date

-Deadline date : Holds date when the rent ends

-Status: Holds the state of the rent, 1 : Pending, 2 : Done, 3 : Late delivery/payment

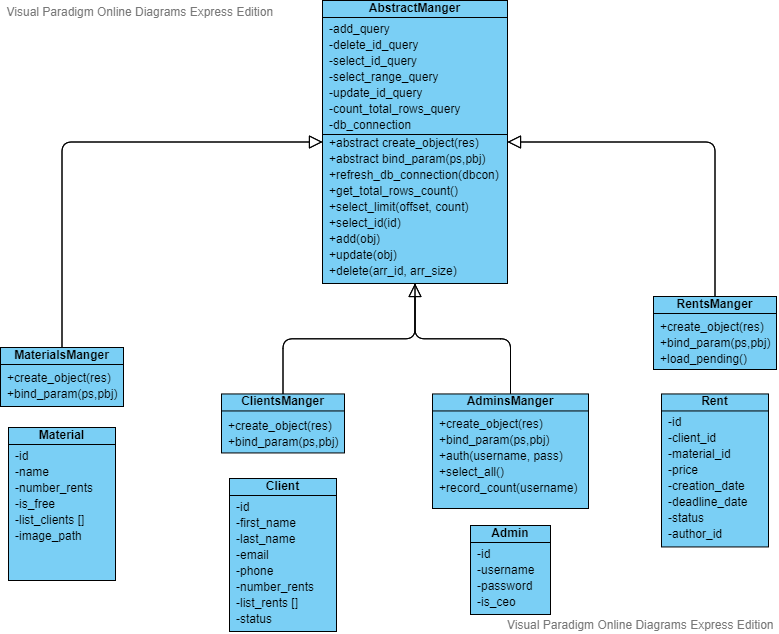
-Author ID: Holds admins id who created the rent

- **Only logged in admin** can Add/Edit/Delete rents

**Building The Core:**

After the planning phase, A flexible backbone was needed to hold the website which allows to extend the website more and add new functionality in the future if needed, a structure have been made to

accomplish what said before as follows :



• core/config.php is website’s configuration file

• Parent class ‘AbstractManger’ attributes must be defined in each child’s constructure.

• Eeach manger must override the abstracted functions

- create\_object(res) : mysqli::get\_result()::fetch\_assoc() will be passed as argument in res , it transfer the raw data from SQL Query to specific object (Client, Admin, ….)

- bind\_param(ps,obj) : prepared statement obj will be pass as ps and the obj that wanted to bind to the statement .

• Each Manger will be created once and save as a SESSION VAR , and must refresh it if needed.

• Manger refreshing (function refresh\_mangers(manger, new\_connection) using bit-flags in systems.php) will refresh db\_connection variable to the new connection because each time page is closed the connection to database is closed too.

• Rents monitor (function monitor\_rents() in systems.php ) will request all pending rents using RentsManger::load\_pending() and check if the rent reached the deadline , if yes it will change rents status 3(Late delivery/payment) and adds one penalty to the client.

• AbstractManger::delete(arr\_id, arr\_size) will delete multiple records passed in arr\_id if arr\_size > 1 , or it will delete only one record if arr\_id is the only one is sent ex: ( delete(5) it will delete the record with id = 5).

• AbstractManger::update(new\_obj) , new\_obj->id must be set , this function helps us to only update wanted attributes without updating all client attributes every time, it will check new\_obj’s attributes and builds UPDATE SQL query for attributes that are set and not null, ex: ( client->id = 5; client->phone = “0665”; client->email = “a@b”; update(client); it will update client’s phone & email with the id = 5 only ) .

• Multi-langauge functionality , when a client visit the website for the first time , a new cookie will be created and set to default language , then language data will be loaded from core/languages.lang and parsed to a key value array and stored in SESSION VAR[LANG\_DATA] .

If the client changed the language , the wanted language will be loaded and then overwrite SESSION VAR[LANG\_DATA] with the new loaded language, the next function can be found in core/config.php .

-LANG(key) : echo value corresponding to the key in SESSION VAR[LANG\_DATA]

-LANG\_1(key) : echo value corresponding to the key in SESSION VAR[LANG\_DATA] but between “ ” ,used in HTML attrbs(href,src,…)

-LANG\_R(key) : Returns raw value corresponding to the key in SESSION VAR[LANG\_DATA] , used in PHP code

**Implementing The Core:**

The core was implemented with the help of HTML Forms and JS, for every form there is onsubmit function to check form inputs, check js/script.js. , Language type is passed to JS on body’s onload function.

• For adding form, all input fields must be set.

• For edit form, only one row must be picked and require at least one field set.

• For deleting, in rent page only one row must be picked, in the other pages picking more than one row is allowed.

**Designing pages layouts:**

• Unified buttons, input fields, titles style.

• Top, bottom bar and side menu have a fixed position.

• Icons are stored in folder named “css”, style datastored in “css/style.css”