Doctor Office

Vianney Portalier

Josselin Araujo

Pierre-Louis Létoquart

Abstract

This project aims to make an interface for a database. Here the database contains information on patient doctors etc. The application communicates with the SQL language. The front-end is implemented in CSS HTML with some parts with the Bootstrap framework. The back end is implemented in php with an MVC architecture for security and easy maintenance. PHP functions allow secure entry of informations before sending SQL requests.

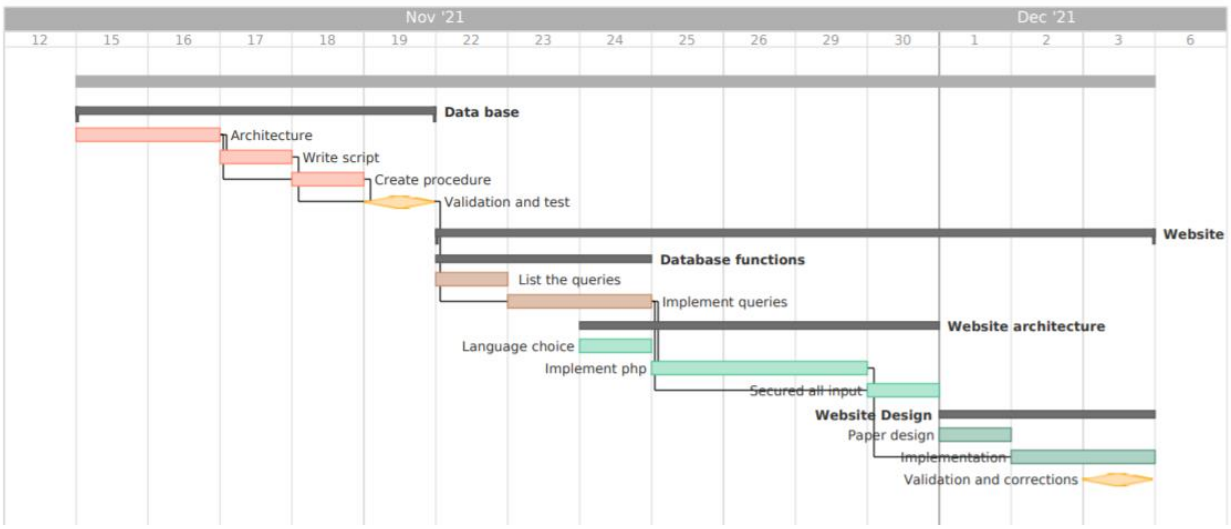
This project was carried out as part of the database course of the University of Ruse (Bulgaria)

Summary

1-Planning

To plan this project, we first defined the list of tasks performed. Secondly, we have grouped these tasks into different groups. Thirdly, we distributed the tasks and defined deadlines for each.

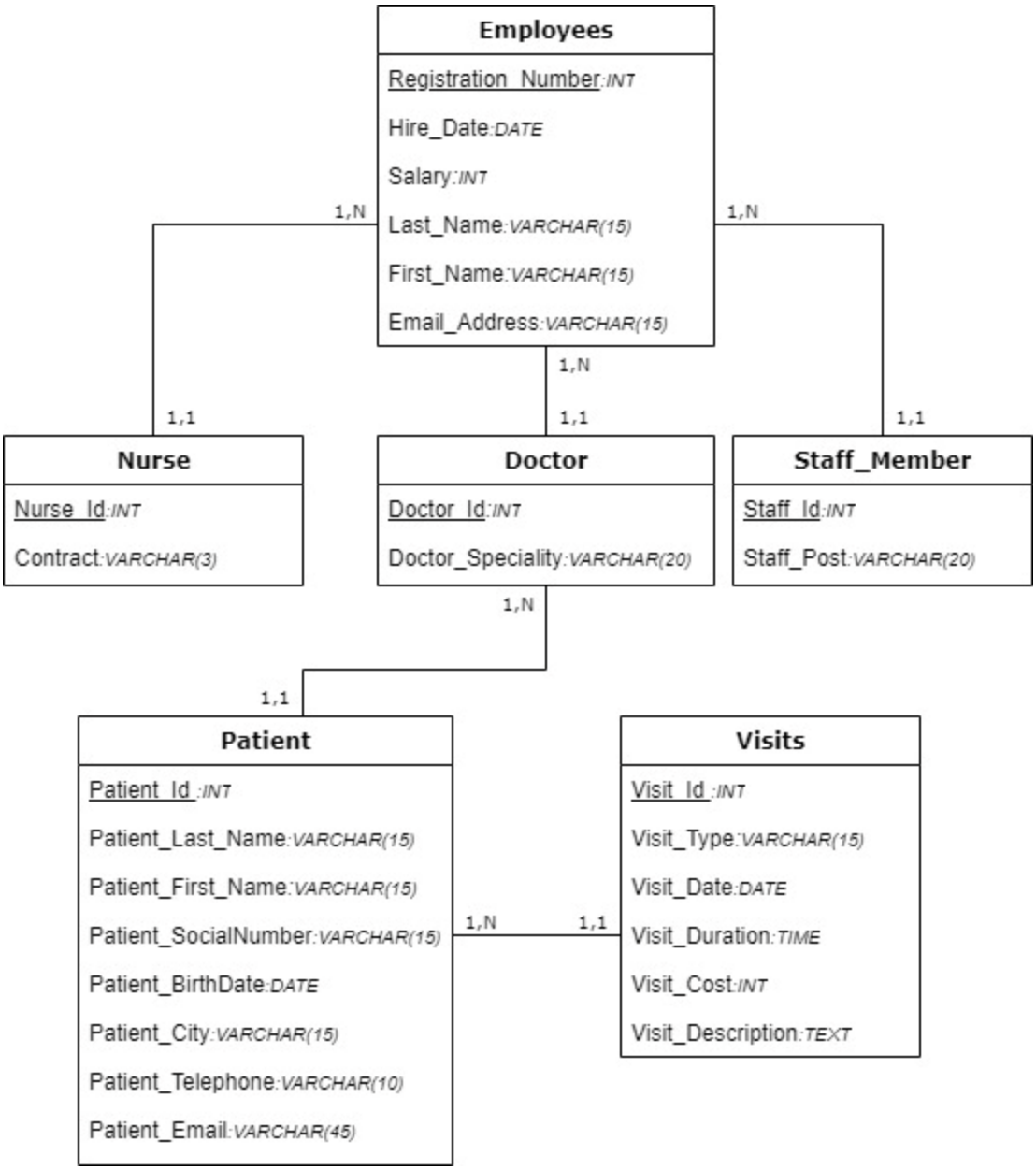
Once the schedule was validated it was put on a Gant chart.



Gant chart - https://www.teamgantt.com/

2-Database

Several database designs have been done. The most complicated being that for a small structure some classes are irrelevant. The architecture was therefore created to be simple but complete.



Database architecture - https://app.diagrams.net/

3-Stored procedures

The stored procedures are not used in the final version of the site. They were created to simplify the reading of the requests and to simplify the operations of the database. No functions were created because they were implemented in php.

```
delimiter &&
create procedure sp_find_regno(in lastname VARCHAR(15), in firstname VARCHAR(15), out var int)
begin
select Registration_Number into var
from Employees
where Last_Name = lastname
and First_Name = firstname;
end &&
delimiter ;

call sp_find_regno('Maman', 'Igor' , @res);
select @res as regno;
```

IDE Vs Code

4-Views

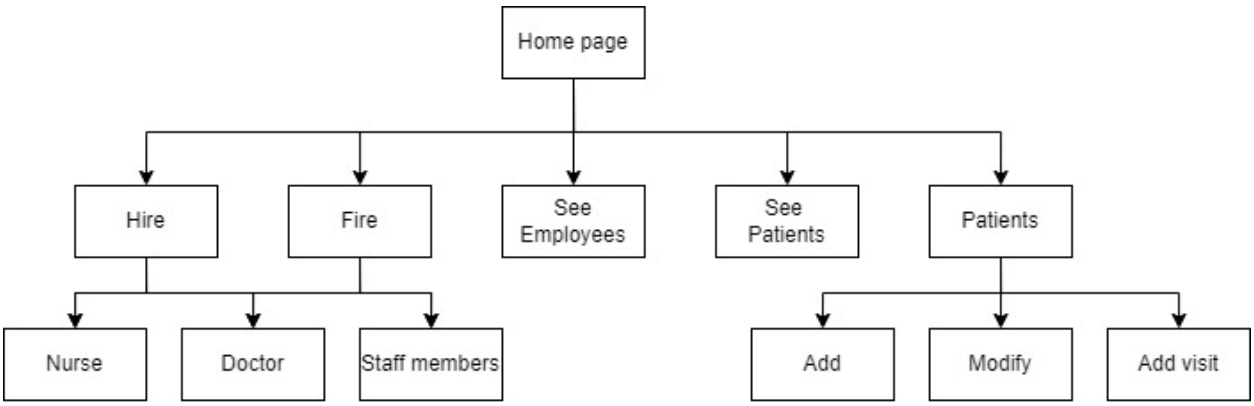
Views were created to simplify testing. They allow you to quickly see if the information has been correctly added or to carry out tests with the information already entered easily.

```
CREATE VIEW [patient_data] AS
SELECT Last_name as docName, Patient_Last_Name as lastname, Patient_First_Name as firstname, Patient_Telephone as telephone
FROM Employees e, Doctor d, Patient p
WHERE e.Registration_Number = d.Registration_Number
AND d.Doctor_Id = p.Doctor_Id;
```

IDE Vs Code

5-Application diagram

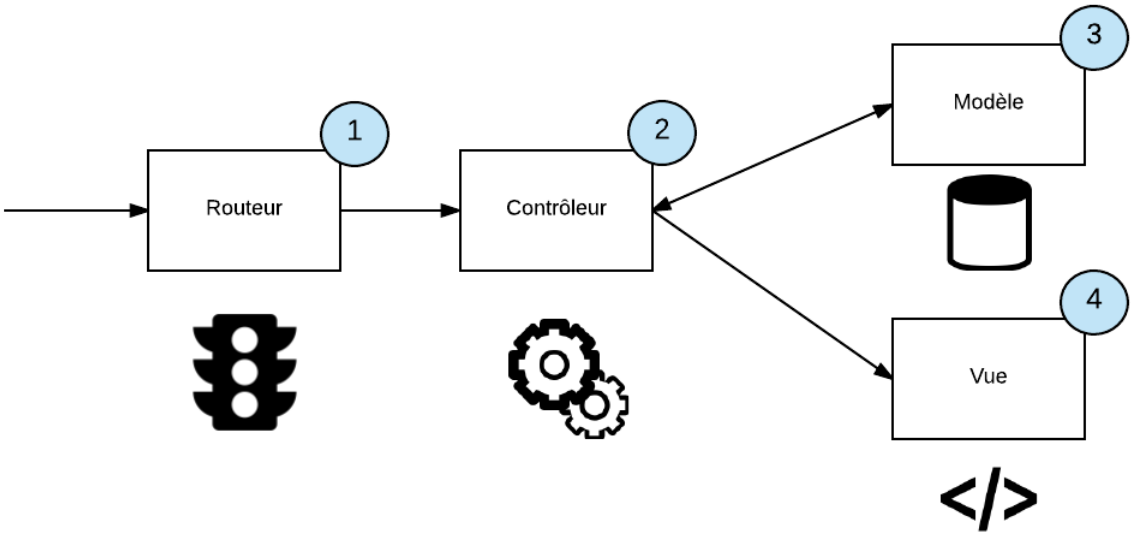
This diagram shows everything that is possible to do from the main page of the site. We find all the basic functions. The functions called are those of the ./view/ folder because they will be if the site is deployed in the public part of the code.



Website architecture - https://app.diagrams.net/

6-MVC Architecture

MVC is an architectural pattern consisting of three parts: **Model**, **View**, **Controller**. Model: Handles data logic. View: It displays the information from the model to the user. Controller: It controls the data flow into a model object and updates the view whenever data changes.



MVC architecture - https://openclassrooms.com/

7-Input security

The security of the entries is made so that the requests do not return errors. This consists in testing the length of the retracted chains. There are also tests performed on certain query responses. For example, if there is no employee with the name ‘xxxxx’ to delete there will be an error message.

```php
if(strlen($socialNo) != 15)
{
    throw new Exception('The social number must be composed of 15 characters');
}

$reg = getDocRegno($docName);
if($reg)
{
    $id = getDocId($reg);
    newPatient($id, $lastname, $firstname, $socialNo, $birthday, $city, $phone, $email);
}
else
{
    throw new Exception('There is no doctor with this name');
}
require('../view/success.php');
```

IDE Vs Code – php code in ./controler/

The use of **try catch** in php makes it possible to not stop the execution of the code while returning a suitable error message which will then be displayed with ./view/error.php .

```html
<div class="input-box">
  <span class="details">Last Name</span>
  <input type="text" name="lastname" placeholder="Enter the last name" maxlength="15" required>
</div>
```

IDE Vs Code – html form in ./view/

In an html form, **maxlength** allows you to control the maximum number of characters in a string. This avoids having to do a lot of php tests afterwards.

8-Sources table

Database

https://www.mysql.com/

https://www.oracle.com/database/

https://www.w3schools.com/

Front-End

https://getbootstrap.com/

https://visme.co/blog/website-color-schemes/

https://developer.mozilla.org/fr/

Back-End

https://www.php.net/manual/

https://openclassrooms.com/

https://stackoverflow.com/

Other

https://fr.wikipedia.org/wiki/

https://app.diagrams.net/

https://www.teamgantt.com/