Messaging App Project

WEB DEVELOPMENT IN SERVER ENVIROMENT
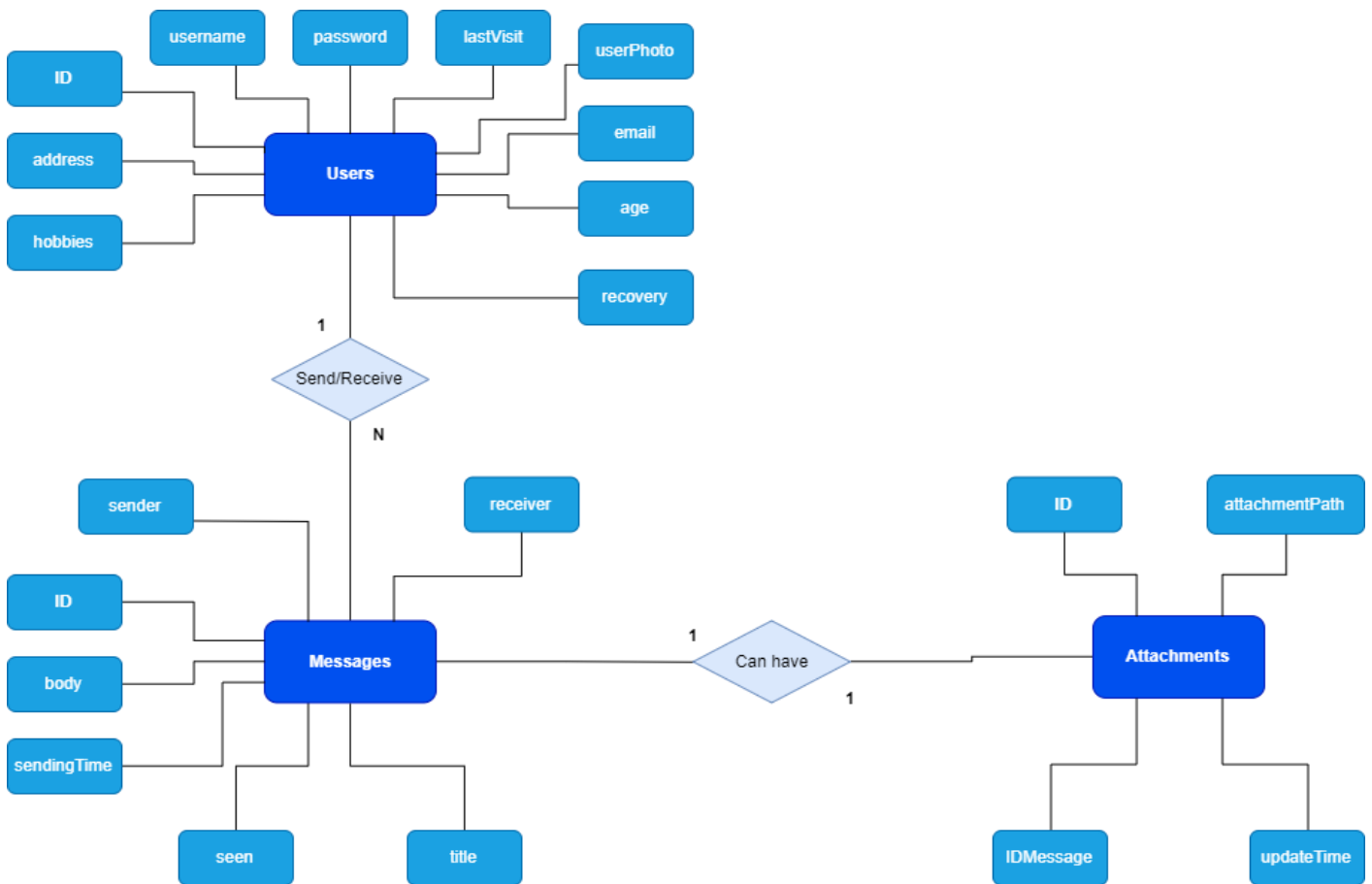
Geroboham Álvarez Hernández
Maria Teresa de Cal Muñoz

DW2E

Requirements specification

EXTENSION	IMPLEMENTED (Y / N)
A1	Y
A2	Y
A3	Y
A4	Y
A5	Y
A6	Y
A7	N
A8	N
A9	N
A10	Y
A11	N
A12	Y
A13	Y

Entity / Relationship Scheme
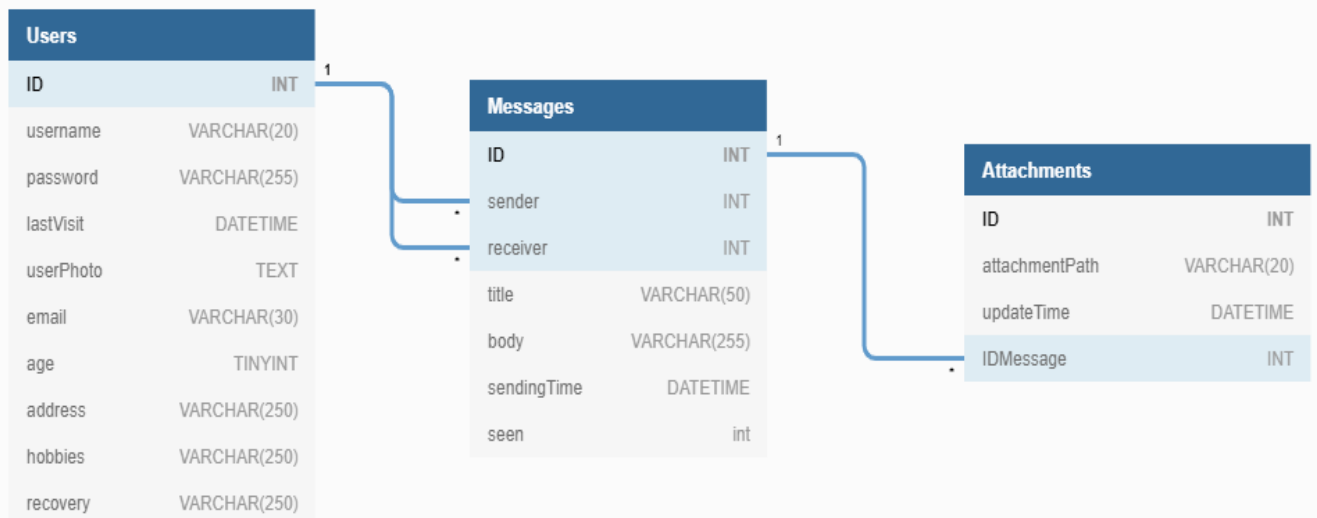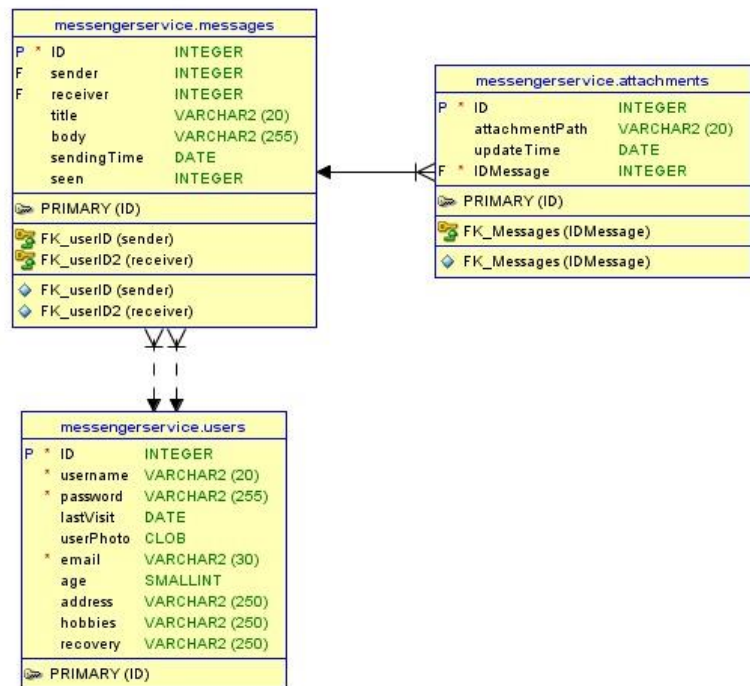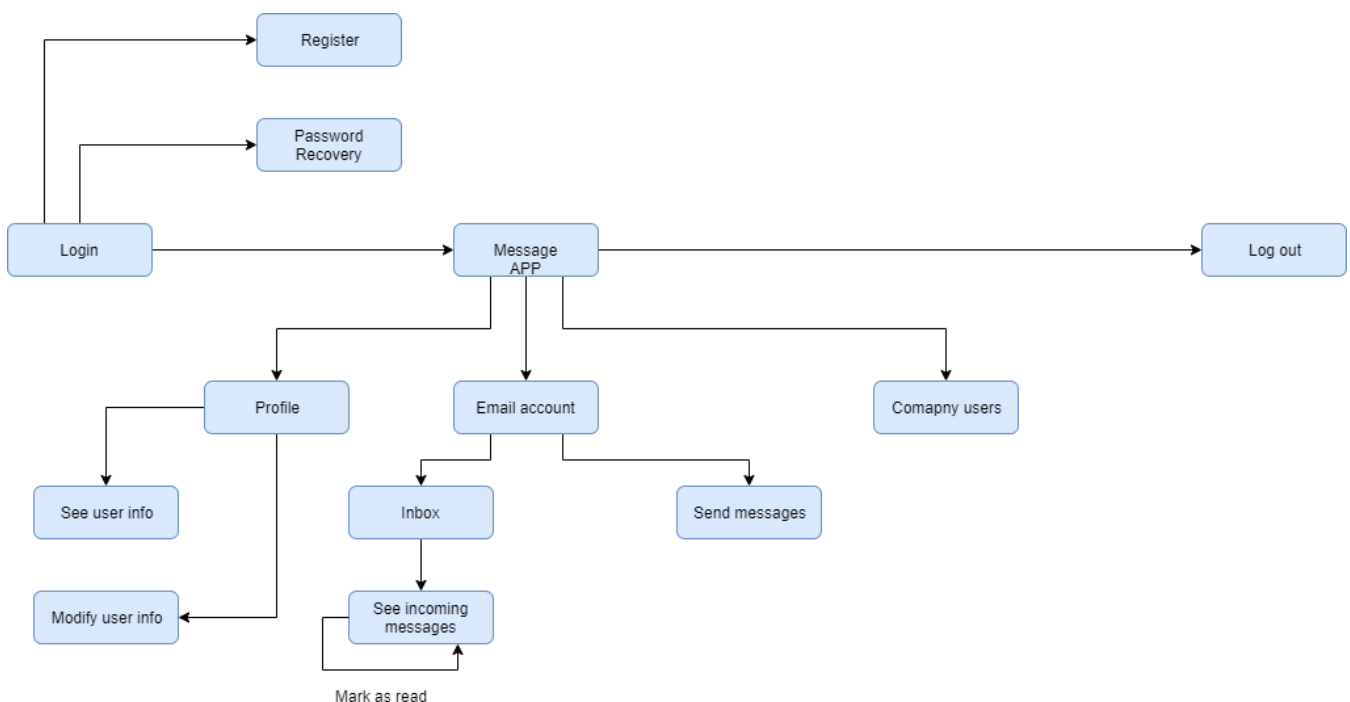

DB Logical Model

Diagram of the database obtained with SQL Developer

Screen Map

Files brief

The base of our proyect are **Entities**, one for each element on the database: **Users, messages and attachment.** Each entity has the same properties as the elements on the database, a constructor and getters and setters to receive the data.

users.php	messages.php	attachment.php
Properties	Properties	Properties
Constructor	Constructor	Constructor
Getters & Setters	Getters & Setters	Getters & Setters

Our app follows the **MVC design pattern**. In our app we have three fundamental parts distributed in different archives:

- **Model:** This layer represents the data. Is the base of the application. The model doesn't depend on the controller or the view.
- **View:** The view display all the data include in model and sends user actions to the contoller.
- **Controller:** The controller provides model data to the view and interprets user actions. The controller depends on the view and the model.

We explain in more detail bellow:

Model

userModel.php
changeUserState($user, $recoveryString)
modifyUsers($user)
modifyPassword($idrec, $password)
updateLastVisit($id)
updateLastVisit($id)
loginUser($user, $password)
getListUsers()
insertUser($user)

messagesModel.php
getMessages($user)
ModifyMessages($message)
sendMessage($message)
getAttachments($msmid)

archiveModel.php
updateAttachments($file)
SaveArchive($file)
AttachArchive($file)

The **Model** contains all the methods that are going to be called on the **Controller**. All the archives has some different methods that principally has to do things with the database directlly and also with directories. In our case, *userModel.php*, *messagesModel.php* and *archiveModel.php* has methods that made a SELECT, INSERT or UPDATE onto the database. Also, *archiveModel.php* has two methods that manage where are located the archives saved on the system.

Controller

userController.php
modifyUsers($user)
modifyPassword($idrec, $password)

messagerController.php
modifyMessage($message)
getMessages($user)
getUsers()
sendMail($message)

archiveController.php
modifyArchive($archive)
attachArchive($archive)
saveArchive($archive)

loginController.php
checkLogin($username, $password)
makeLogin($user)

recoveryController.php
sendRecoveryMail($user, $address)

The **Controller** as we said before is in charge of call the functions in **Model** and navigates throught them. We have one for each entity, *userController.php*, *messagerController.php*, *archiveController.php* and two new, for manage the Login on the app, *loginController.php* and for manage Password recovery in case you forgot your password, *recoveryController.php*. This files also call functions found in the model.

View

The **View** is responsible for displaying everything that is generated in both **Model** and **Controller**. The view is shown onto a web browser and created a nice look from the data.

We have *.php* files that mostly has html code, to show the data on the browser. This files, contain different calls to js files and php files.

In view folder we have 4 subfolders: **attachments**, **css**, **img/profile_photo** and **js**:

- In *attachmets* , uploaded files are saved when sending a message. When stored in this folder, for the user who receives the file can download it.
- In *css* we have the external css for the view of the page. We use Sematic UI framework.
- In *img/profile_photo* we storage the profile photos that the users upload to our app.
- In *js* we have external js files that contains different functions and methods that appear on the view, as dinamic elements as click a button, disable buttons, create tables, and the principal core, receive **AJAX requests**. Ajax request are very important in the 'main body' of our app, our mail service, that entirely is made on AJAX.

AJAX:

As we said, our mail service is managed through AJAX request, so we need to explain more in detail the different folders and files that are called to manage the requests.

Outside the view, we have a folder called **ajax**, that contains different *.php* files with functions that manage conections between server side and view. This is the base for the AJAX request, that we made in *httpModule.js*. In this file we create the functions that made the requests.

This file is called on messager.php and here it happen the connections between server and client.

View files

In the table below, we specify the description, the parameters and the redirect to each file.

Path	Description	Parameters	Redirects to
login.php	Login form	$_SERVER['REQUEST_METHOD'] $_POST['username'] $_POST['password']	messager.php register.php forgotpassword.php login.php
register.php	Register form	$_SERVER['REQUEST_METHOD'] $_POST['username'] $_POST['password'] $_POST['email']	messager.php login.php
forgotpassword.php	Password recovery form to recieve a email for change the password	$_SERVER['REQUEST_METHOD'] $_POST['username'] $_POST['email']	login.php forgotpassword.php changepassword.php?changeid=…
changepassword.php	Password recovery form	$_SERVER['REQUEST_METHOD'] $_POST['username'] $_POST['newpassword'] $_POST['newpassword2'] $_GET['changeid']	login.php
messager.php	Email inbox	$_SESSION['user']	profile.php messager.php companyUsers.php login.php

profile.php	User avatar and info	$_SERVER['REQUEST_METHOD'] $_FILES['PhotoProfile'] $_SESSION['user'] $_POST['age'] $_POST['address'] $_POST['Hobbies']	profile.php messager.php companyUsers.php login.php
companyUsers.php	List of all the users registered on the app	$_SESSION['user']	profile.php messager.php companyUsers.php login.php

User's guide

Introduction

This app has being designed for use as a business email application. It has been programmed for use through XAMPP, using Apache and PHPMyadmin databases. It can be run on any Windows OS that as this tool installed (We recomend Windows 10). For the use of some functions it is necessary that the user owns a valid and fuctional email address.

To 'launch' the program, just go to *htdocs/MessengerServices/src/views* and open ***login.php*** file.

APP Requirements

For the correct performance of the program it is necessary to have a series of requirements:

- Last version of XAMPP

- Star Apache and MySQL Modules on XAMPP

- Last version of a web browser (Recommended Google Chrome or Mozilla)

- Charge the database of the APP in phpMyAdmin: Opening a web browser and typing localhost/phpmyadmin. First, you need to go to SQL tab and copy the content in ***setup.sql***, available on *htdocs/MessengerServices/src/config*. Now that the database is created, you need to copy the content in ***messengerserviceDB.sql***, available on the same folder mentioned before, and repeat the steps bellow. Once you click on 'Continue' button, the APP is ready to start.

- We use PHPMailer in one function in the APP. In order to check if this function works, is needed to configure the file **recoveryController.php**. You can found it in *htdocs/MessengerServices/src/controller*. You need to add your mail configuration in line 27, 28 and 29.

    IMPORTANT! Your email provider MUST BE Gmail. Here's an example.

```
- $mail->Username = "example@gmail.com"; // Here goes your email
  $mail->Password = "password"; // Here goes your email password
  $mail->SetFrom("example@gmail.com", 'NoReply' ); // Here goes your email
```

Logging in

The first screen you seen when you 'start' the application is the **Log in** with a **User** and a **Password**.

Log In

User *

Username

Please enter a value!

Password *

Password

Please enter a valid password!

Forgot password?

| Log In | Register |

We have some test data to get Log in, that are the following:

User: Mayte   **Password:** Mayte1234

User: Gero   **Password:** Gero1234

User: Test   **Password:** Test1234

You have to enter the credentials and if all the data is correct the button **Log in** will be activated to get into the application.

Register

On the **Log in** page appears a second button, that guides you to another screen to **Register** onto your application. It will appear another form with **User**, **Password** and **Email**.

In here we have some specifications:

- The **user** can't have blankspaces, must be a single word.

- The **password** should contain at least one digit, one lower case, one upper case and 8 characters from the mentioned previously.

- The **email** should be a valid email. This is important because is the email where the password recovery will be sent in case the user ask for it.

Once you enter all the form fields and if all the data is correct, the **Register** button wil be activated and you get into the application.

Register

User *

> Username

Please enter a valid username!

Password *

> Password

Please enter a valid password!

Email *

> Email

Please enter a valid email!

Register	Reset	Log In

Recovering your password

Password Recovery

User *

> Username

Email *

> Email

Email me	Reset	Log In

In case you register on your app and you aren't been able to remember your password, you can recover it. You just press '**Forgot your password**' link and fill the form. We'll send you an email with a link in which you can change it.

Password recovery Recibidos ×

NoReply <phpserverphp@gmail.com>
para mí ▾

文A inglés ▾ > español ▾ Traducir mensaje

Hello, Mayte

Forgot Your Password?

It happens. Click the link below to reset your password.

Reset password

You're getting this email because you've signed up for email updates.

Change your password

User: *

> Username

New password: *

> Password

Repeat password : *

> Repate Password

| Submit | Reset |

Once you change your password, the button submit guides you to the **Log in** form, to get again in the application.

Your profile

As user of our application you have a **profile** with your data. At the beginning it only appear your name and email that you put when did you register. You can change this data, in the next step we will see how you can do it. When you change the data, it will appear in your user card.

Welcome TEST   Profile   E-mail account   Company users
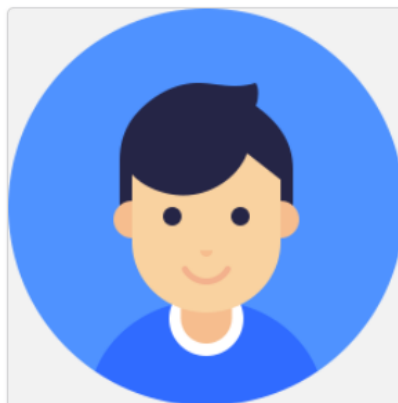
Test
test@test.com
Last time visit 2019-11-30 09:11:46
Age
Home Address
Hobbies

MODIFY

Mayte
mcalmunoz@gmail.com
Last time visit 2019-11-30 09:11:52
Age 28
Home Address Madrid
Hobbies Listen to music

MODIFY

Updating your profile

To **update** your data on the profile, you just need to press **Modify** button. It will appear a form where you can introduce the different data you want to add. It is no necessary to fill all, you can only upload a photo or just add your age, address or hobbies.

Email account

This is the **main** of the application, where you can send and receive mails between the users registered previously in the application.

Your Inbox

Here it will appear the **messages** you receive, differentiating between read and unread messages.

When you click on a message, it will apear on the right to read it. If it isnt read, a check will be put marking as seen.

Now, you can response or download the files attached on the message if there's any. If there's a file you can download, it will be saved on your pc; if isn't, a message tells you that the're no files to download.

If you want to response the message, you click on the response button, that will take you to **Send Messages** tab.

Sending mails

Sending to one user

Here's where you can send messages to the users registered. It will apear a list of the users availables. If you want to send it to one, just select it with the mouse. Then, fill the form with the title, your message and attach a file if you want, it is no necessary, you can send a message without attach file as we said before.

Sending to multiple users

If you want to send the same message to two different people, select the first as before and to select another, push 'CTRL' on the keyboard and then click in the user you want.

Responding a message

If you click on response, you just need to fill the body of the message, the user receiver of the message will be selected already, because you're responding to a message that this user has sent you, and also the title, will be 'RE: *title of the message*'.

Company users

In here you will find different informative cards about the users registered. You can see the data that the different users filled in his/her profile. It's just informative to know better your partners.

Logging out

To finish using the mail service, it is as simple as pressing the **Log out** button at the top right of the screen. This will close our session and we will be finished.