First Term Proyect

Adrián Rodríguez Fernández
Daniel Gil Fernández
Madrid
24/02/2021

INDEX

1. REQUIEREMENTS SPECIFICATION

You must include a table that indicates which extensions have been made.

We want to make a chat application. Users can use the application to send texts and files to other users.
The application allow the user to:
- Register and login with cypher password.
- Send messages to other user.
- Send the same message to several users.
- Receives messages.
- Create a friendship with other user. User have to accept the friendship.
- See profiles from other users if both users are friends.
- Modify the user profile with a custom avatar.
- Create a group with several user and see the conversation between them and participate.

EXTENSION	IMPLEMENTED
Self-registration	YES
Password recovery	NO
Messages to several recipients	YES
Encrypted user password	YES
User avatar	YES
User profile	YES
Friendship	YES
Groups	YES
Administration zone	YES
Attached files	YES
Images	YES
AJAX	YES
Presentation	YES

Figure 1: Extensions Table

2. E/R SCHEME

We have 3 entities, the user, the room and the message.
The user´s attributes are related to its personal information such us name, mail, password which is encrypted.

The users are in rooms, this rooms can be of two types: direct messages (from one user to another) or in a group (from one to more user). Each group has its own image that shows in each participant chat list. And the room has a code that identifies it, for direct messages the code is the participants code order from lower to highest. And if it is a group the code is the name of the group.

Each room is composed of messages. They have the code from the user that wrote it, the text that can be either a text or a file and the date when the message has been send.
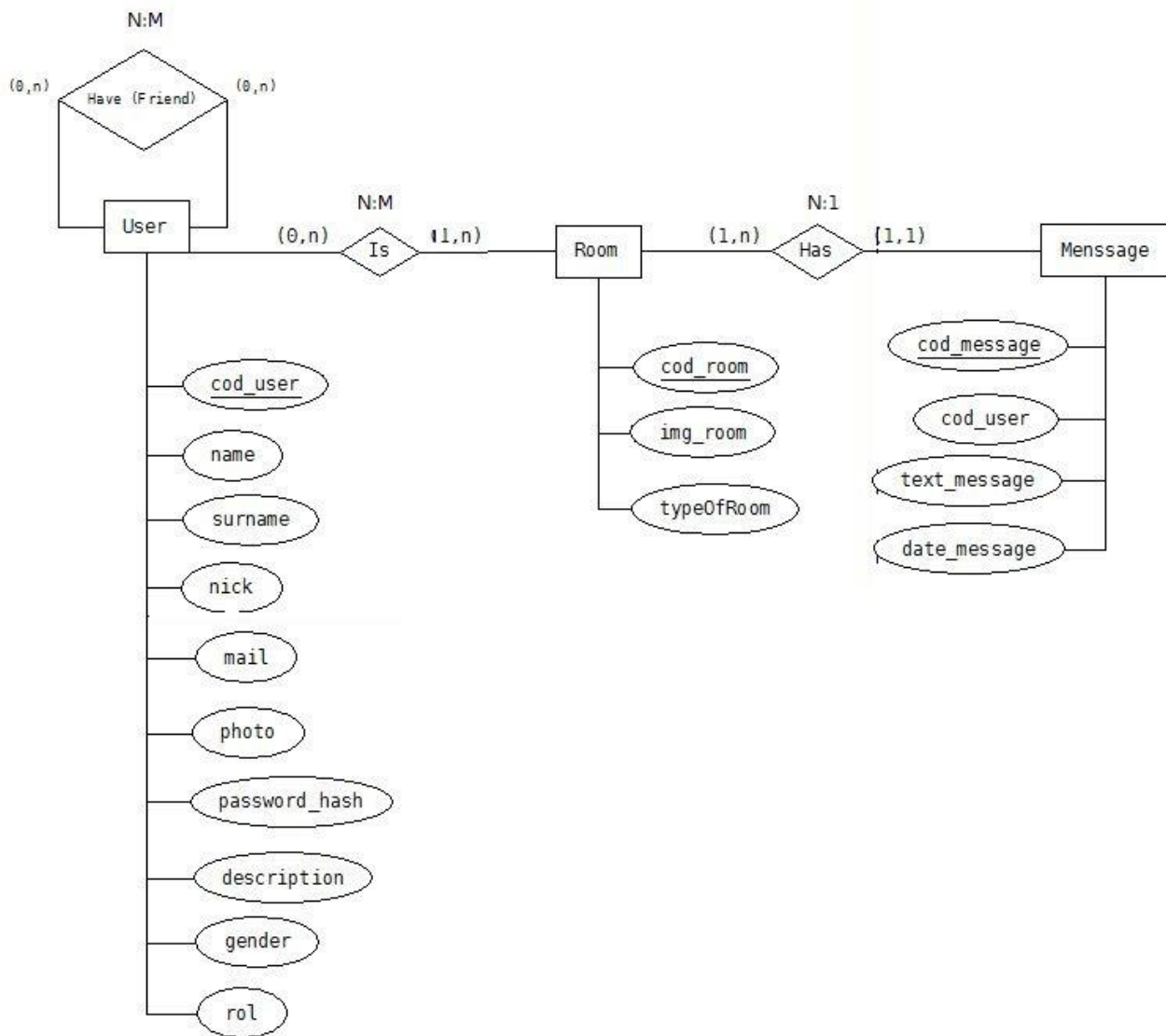
Figure 2: Entity Relationship Diagram

3. DIAGRAM OF THE DATABASE OBTAINED WITH SQL DEVELOPER

For the friendship system another table had to be made, this table contains both users, a status that indicates if the petition has been accepted or not and a code with both users codes from lower to highest.

Also we needed a table between user and room as a user can be in one or more rooms and the room can have one or more users, inside this table we have the field view. This field allows a user to see if a chat has a new message since it was last opened.
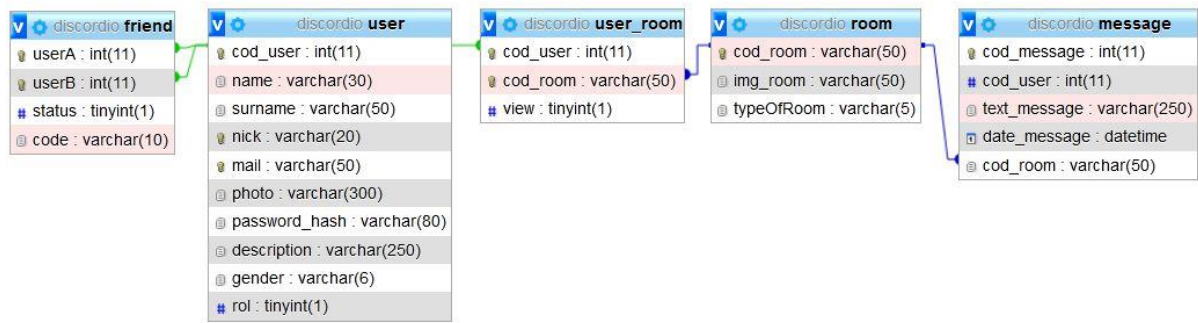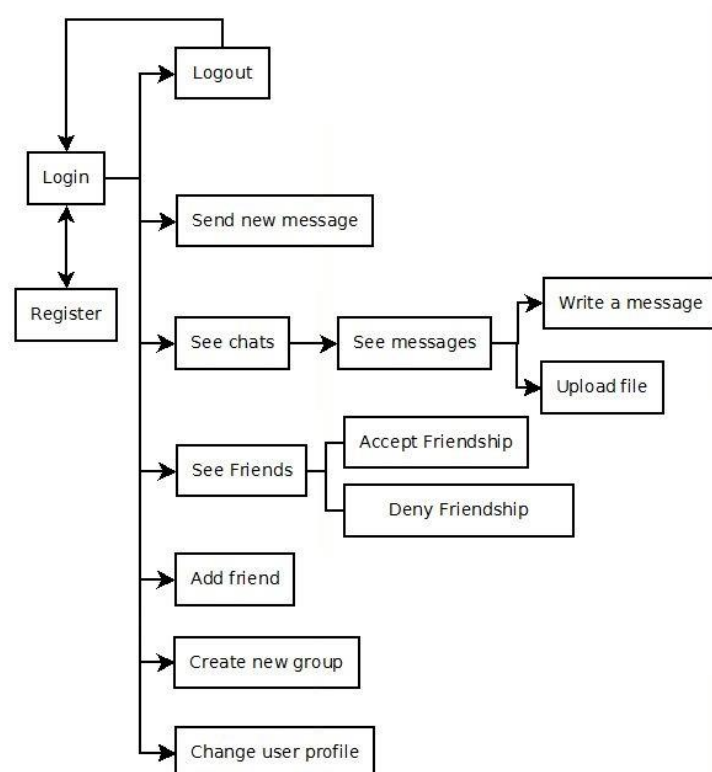
Figure 3: Diagram of the database

4. SCREEN MAP

Figure 4: Screen Map

5. SUMMARY TABLE OF THE FILES THAT ARE PART OF THE APPLICATION

Path	Description	Parameters	Redirects / Calls to
/index	Initialises the variables of codUser and rol		Index.html.twig
/main	Main page		Main.html.twig
/acceptFriend	Is used for accepting a friend(The tick in the petition)	$_SESSION ['codUser'] $_POST['codUser']	index.html.twig
allUser	Loads all the profiles for the admin to see		allUser.html.twig
chat_AJAX	Loads the chat with users or groups and allows to send messages	$_POST['codRoom'] $_POST['avatar_chat'] $_POST['name_chat'] $_SESSION['codUser']	chat_AJAX.html.twig
contacts	The php for the left side of the screen	$_SESSION['codUser']	contacts.html.twig
createNewGroup	Creates a new group	$_POST['users'] $_POST['nameGroup'] $_SESSION['codUser']	index.html.twig
denyFriend	The php for denying a friend(The cross in the petition)	$_POST['codUser'] $_SESSION['codUser']	index.html.twig
Friend	Shows the friends of the user	$_SESSION['codUser']	friend.html.twig
Logout	Logs out of		index.html.twig

	your sesion		
newGroup	Button for creating a new group		newGroup.html.twig
profFriend	Allows to see the profile of your friend	$_POST['codUser']	profFriend.html.twig
profile	Allows to change the user profile and also is where you add your picture and description	$_SESSION ['codUser']	profile.html.twig
request_friend	The request of friendship from one user to another	$_POST['user'] $_POST['text'] $_SESSION['codUser']	index.html.twig
search_bar_friend	Search for looking for a new friend		
search_bar	Search for looking for a user to send a new message		Search_bar.html.twig
send_message_AJAX	Allows to write messages from one user to another	$_POST['codRoom'] $_POST['text'] $_SESSION['codUser']	index.html.twig
send_message_newMessage	Sends a new message to a user	$_POST['codRoom'] $_POST['text'] $_SESSION['codUser']	index.html.twig
send	Where the message is actually	$_POST['user'] $_POST['text'] $_SESSION ['codUser']	index.html.twig

	sent		
sendFriendship	Sends the request of friendship from one user to another	$_POST['nameUser'] $_SESSION ['codUser']	index.html.twig

upload_files	Allows to upload files in the chat	$_POST['codRoom'] $_POST['cod_my_usr'] $_POST['code_room'] File→get('file') $_SESSION ['codUser']	index.html.twig
uploadProfile	Updates the changes made to the user profile	$_POST['name'] $_POST['surname'] $_POST['description'] $_POST['nick'] File→get('file')	index.html.twig
login	Login screen		security/login.html.twig
register	Creates your user if you don't have one	$_POST['name'] $_POST['surname'] $_POST['nick'] $_POST['email'] $_POST['password'] $_POST['gender']	index.html.twig

/public/functions.js	Here are all the functions that call the php for the main page		

6. DOCTRINE EXAMPLES

```
$codUser=$request->get("codUser");
$entityManager = $this->getDoctrine()->getManager();
$user = $entityManager->find(User::class, $codUser);


$entityManager->remove($user);
$entityManager->flush();
return $this->render('index.html.twig');
```

Figure 5 Delete

```
$query = $entityManager->createQuery(
    'select u from App\Entity\User u WHERE u.codUser like :coduser'
)->setParameter('coduser', $coduser);
```

Figure 6 Select

```
$new = $entityManager->find(User::class, $cod);

$new->setName($name);
$new->setSurname($surname);
$new->setPhoto($photo);
$new->setDescription($description);

$entityManager->persist($new);
$entityManager->flush();
```

Figure 7 Update

```
$password_hash = password_hash($password, PASSWORD_DEFAULT);

$new = new User();
$new->setName($name);
$new->setSurname($surname);
$new->setNick($nick);
$new->setMail($email);
$new->setPhoto('default.png');
$new->setPasswordHash($password_hash);
$new->setDescription('');
$new->setGender($gender);
$new->setRol(0);

$entityManager->persist($new);
$entityManager->flush();
```

Figure 8 Insert

7. USER MANUAL
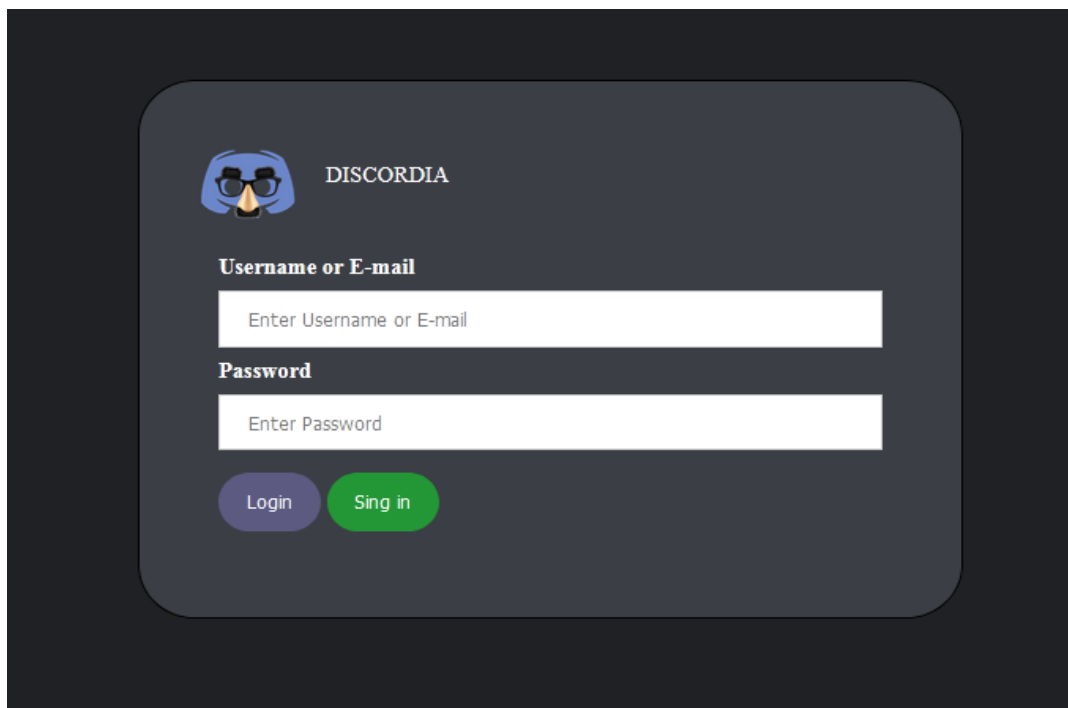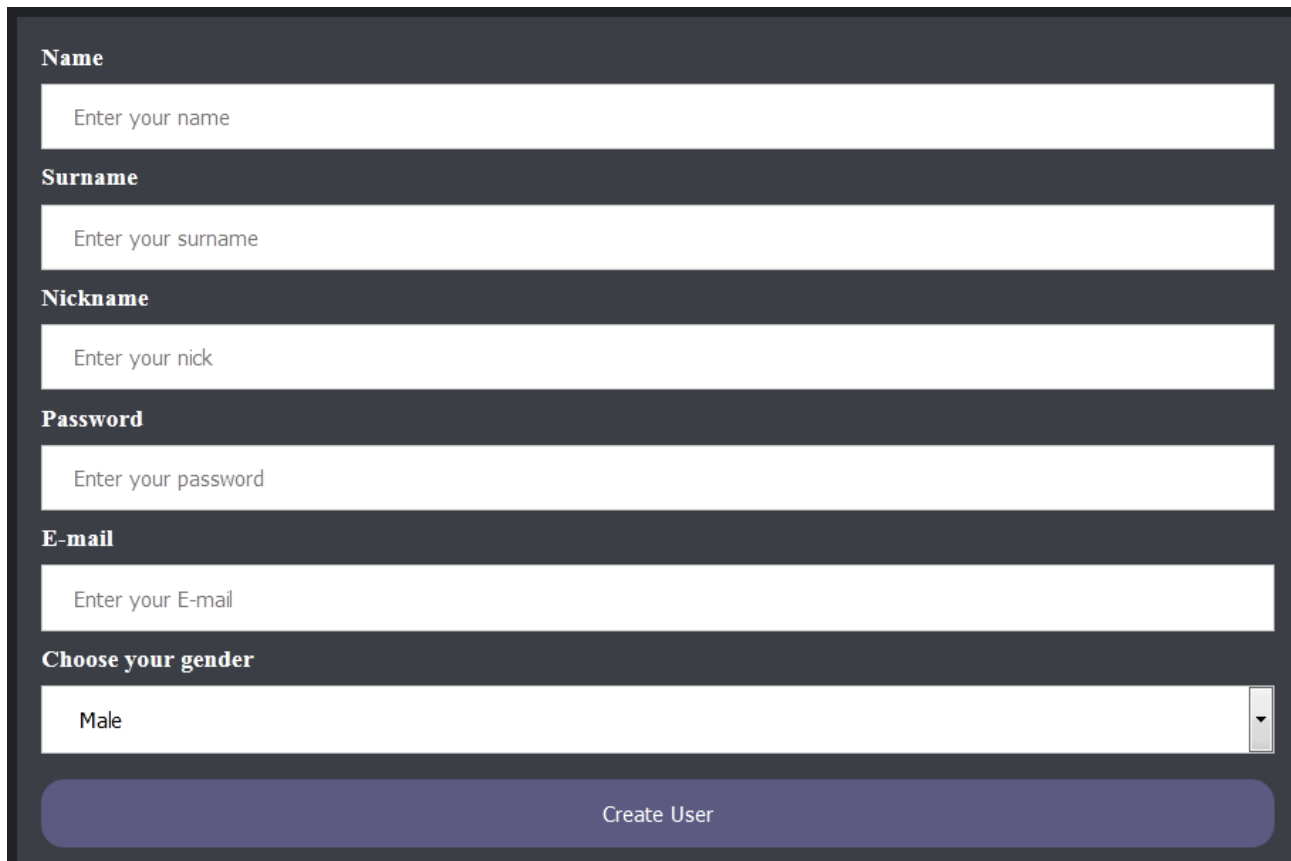
Figure 9: Login Form

This is the login form. You can login with your nick or email if you have an account or create a new profile

Figure 10: Register Form

This is the register form where you introduce your data for the new profile.
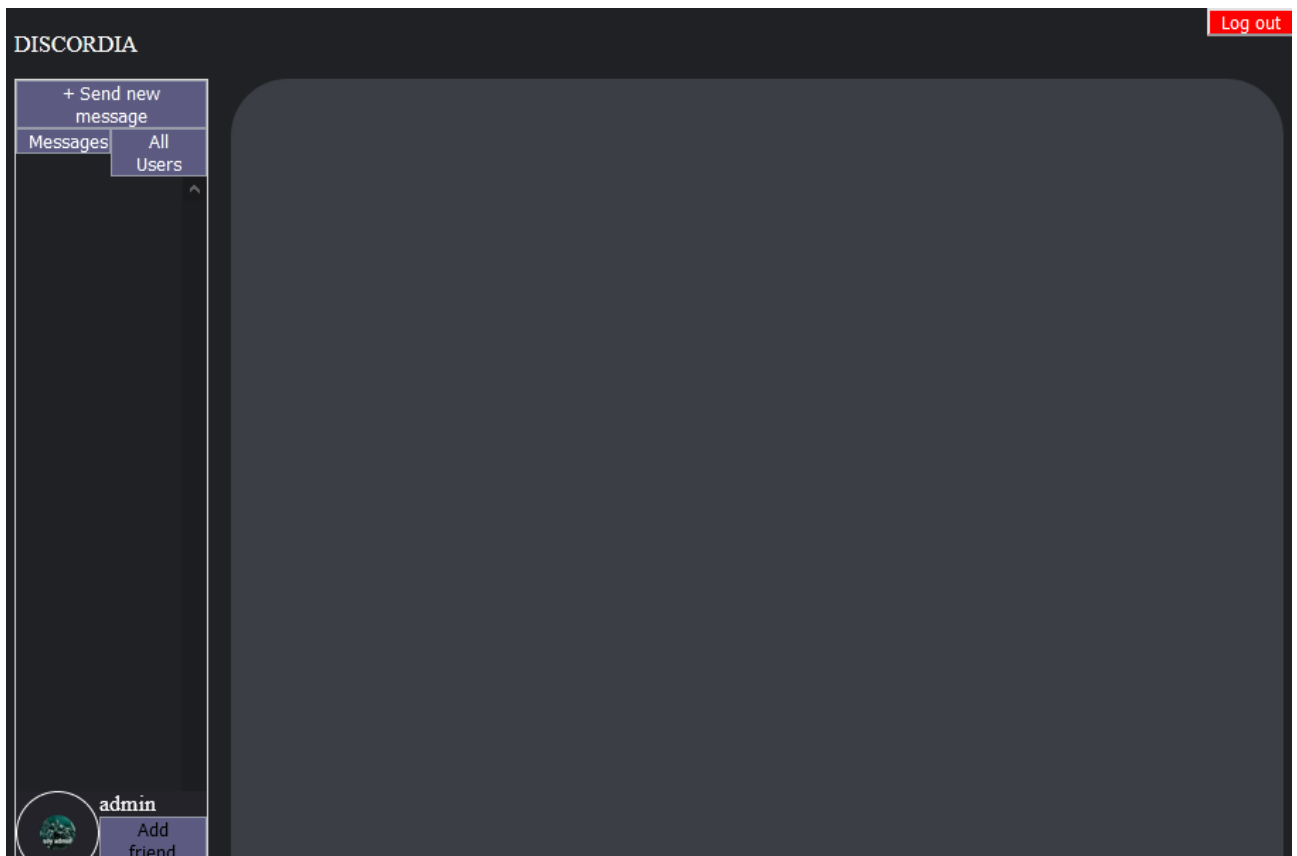
Figure 11: Home Page

This is the admin profile where as an admin he can see all the profile that are store in the data base beside doing all the other functions a regular user could do.
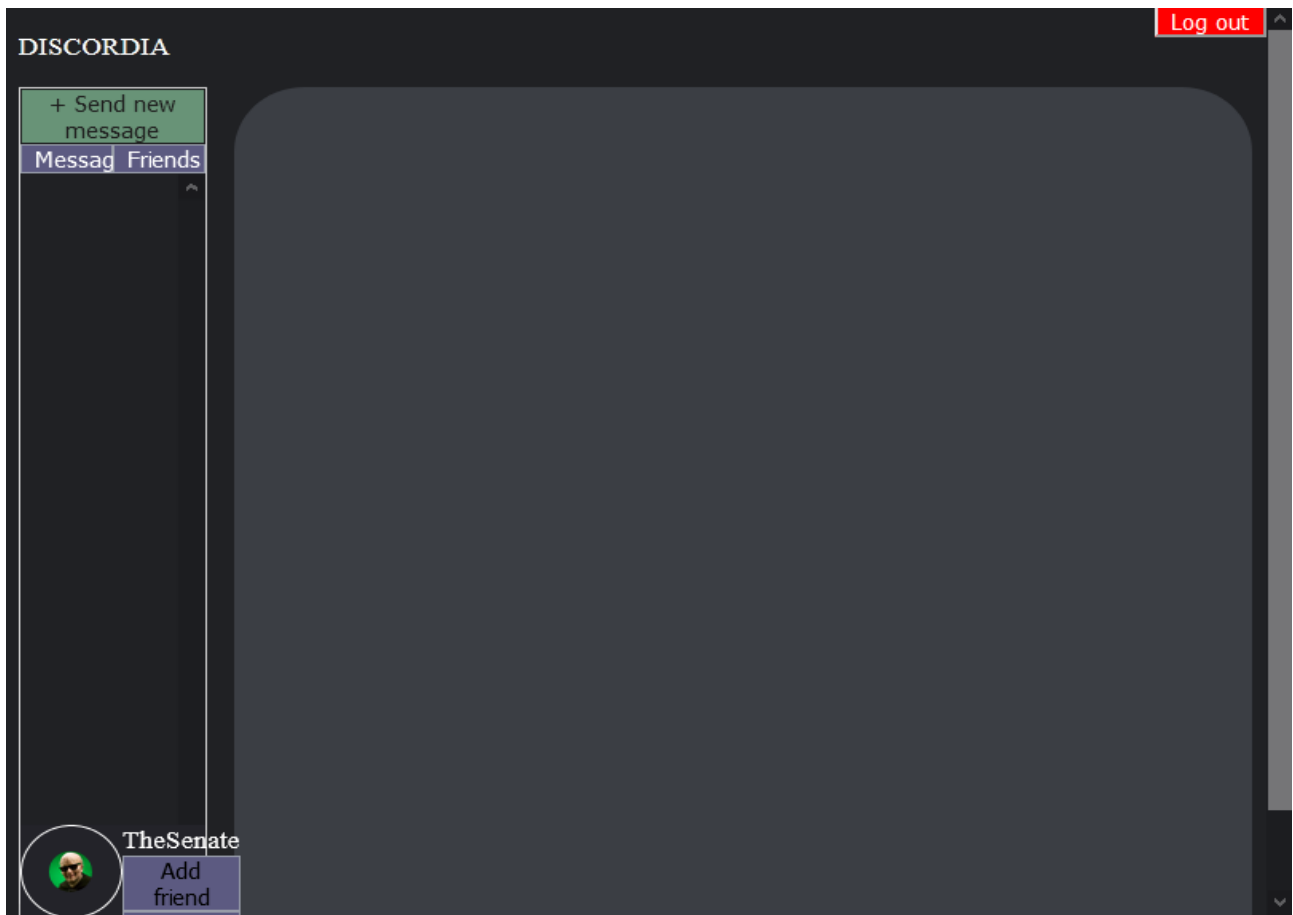
This is a regular user main page. Where it can see its messages, add friends, send new messages, create groups or logout.

Figure 12: Send New Message From

This is the form for a new message where you can write the user or users (separated by spaces) and the text you want to send them.
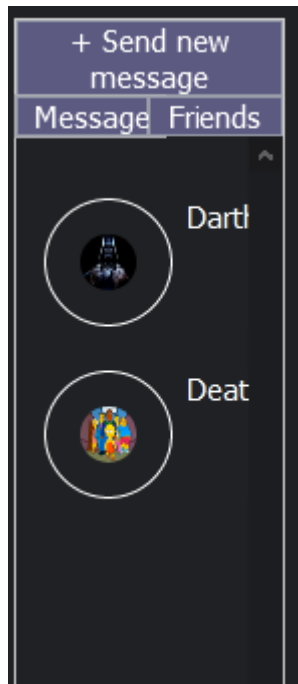
Figure 13: Message Chats Actives

This is the space where the messages from users and groups are seen. If the icon is clicked, it will show the conversation.
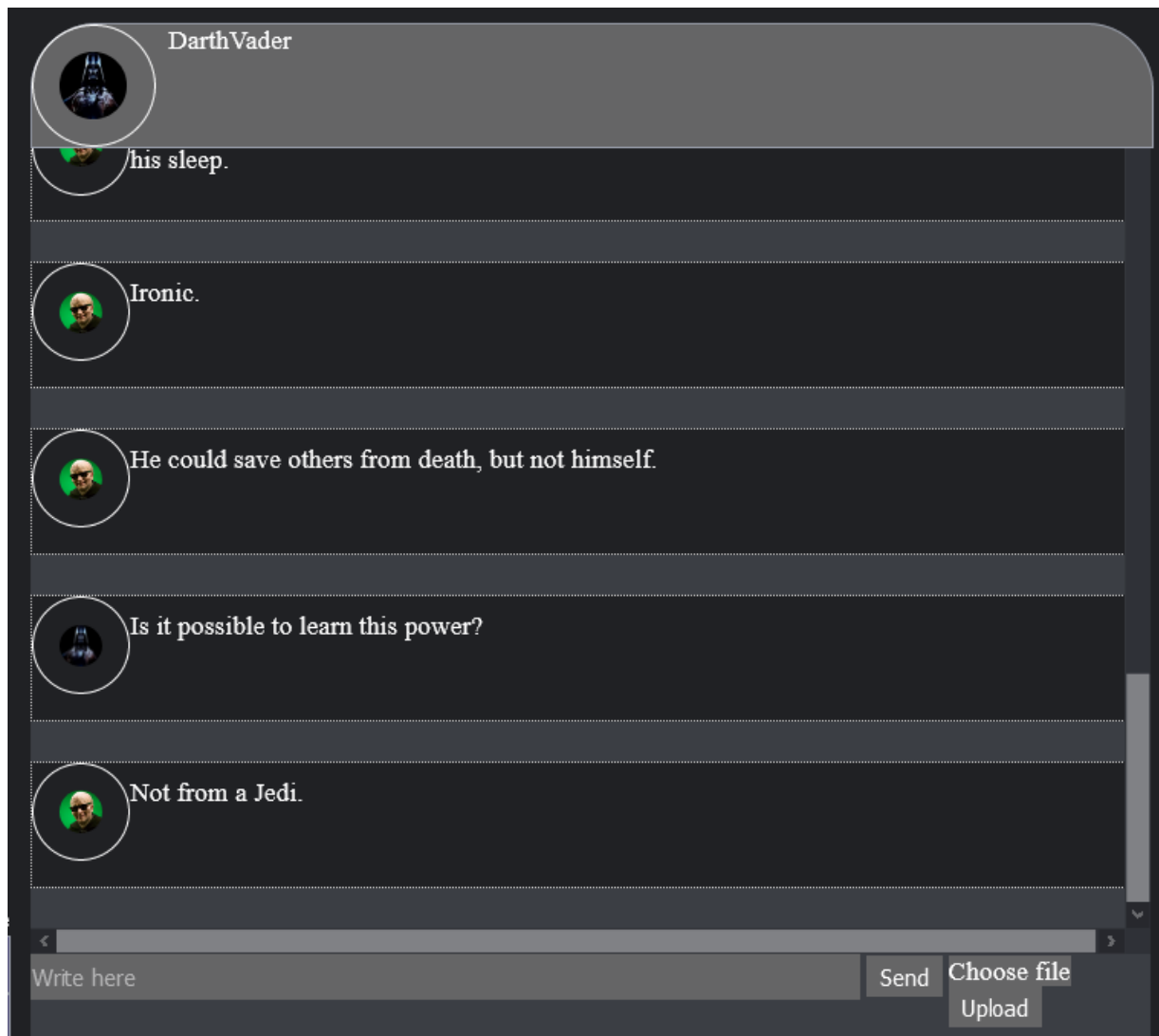
Figure 14: Message Chat Content

This is the chat where the user can read or write messages or send files.
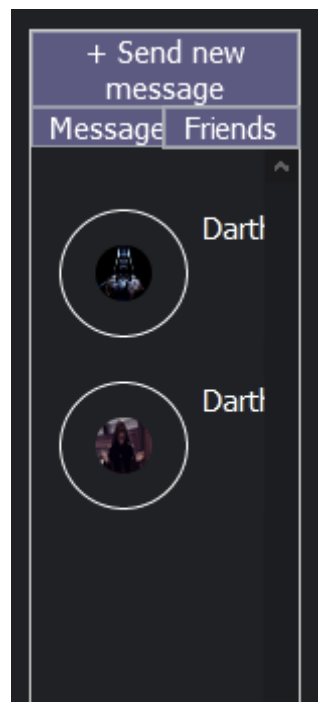
Figure 15: Friend Section
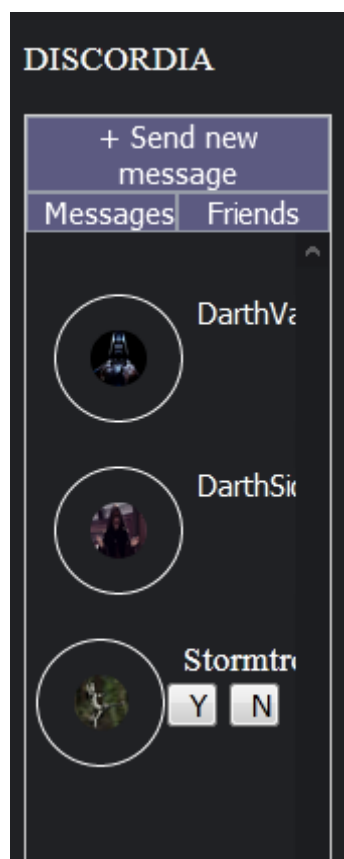
This is the space where the friends from the user are seen. If the icon is clicked, it will show the profile of your friend. If you have a request you can either accept it or deny it.
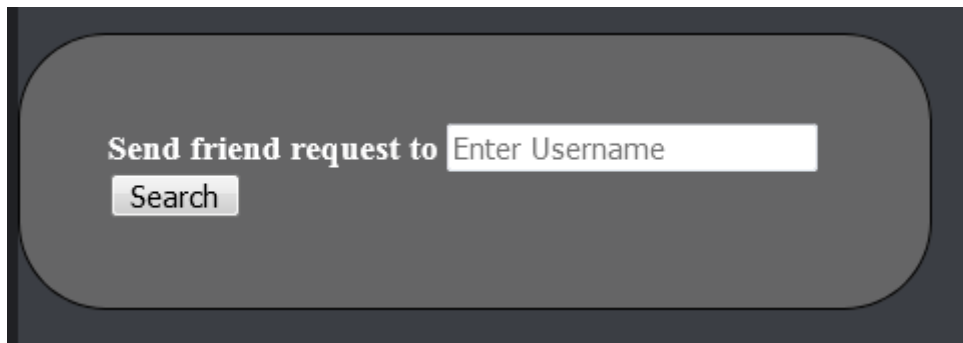
Figure 16: Send Friend Request Form

This is where you search a new friend by entering its nick.
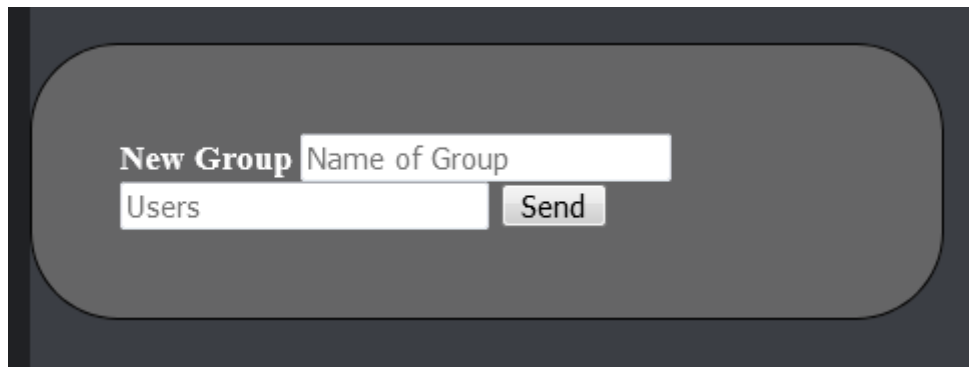
Figure 17: Create New Group Form

This is where you create a new group by entering the group name, and the users it will contain, by default the first message of a group will be from the user that created it.

Figure 18: Profile User Section

This is your profile section, where you can modify your personal information and also sends your default profile picture and add your description.