

Diario di lavoro

Luogo	Arti Mestieri Trevano
Data	26.11.2019

Lavori svolti

Oggi ho gestito il fatto che l'utente può dimenticare l'email, quindi sotto gli input del login ce un bottone "ho dimenticato la password" e quando viene cliccato compare un prompt che richiede l'email a cui si vuole cambiare la password, viene eseguito il controllo dell'email e se è corretta e già presente nel database viene generata una nuova password di default che poi viene impostata nel database e viene inviata per email all'utente.

Quando viene schiacciato il bottone viene fatta partire una funzione javascript:

```
function changePassword(){
    var emailToChange = prompt("Perfavore inserisci la tua email:");
    if(emailToChange.length > 0) {
        window.location.href = URL+"login/updatePassword/"+emailToChange;
    }else{
        alert("Devi inserire per forza un email");
    }
}
```

Che poi porta al controller del login per l'update della password

```
public function updatePassword($email){
    $newPasswordGenerate = $this->userModel->generetaRandomPassword();
    $passwordHash = password_hash($newPasswordGenerate, PASSWORD_DEFAULT);
    if($this->userValidation->validateEmail($email)){
        $this->userModel->updatePassword($email, $passwordHash);
        $this->mailModel->sendEmailUpdatePassword($email, $newPasswordGenerate);
        $_SESSION["errorRequestNewPassword"] = 1;
    }else{
        $_SESSION["errorRequestNewPassword"] = 2;
    }
    header("Location:".URL."login");
}
```

Nello user model ho aggiunto questo metodo che mi fa un update della password e del flag per il cambio password

```
public function updatePassword($email,$password){
    $changePassword = 0;
    $updatePassword = "UPDATE utente SET cambioPassword=:cambioPassword,
password=:password WHERE (email=:email)";
    $this->statement = $this->connAccess->prepare($updatePassword);
    $this->statement->bindParam(':cambioPassword', $changePassword, PDO::PARAM_STR);
    $this->statement->bindParam(':password', $password , PDO::PARAM_STR);
    $this->statement->bindParam(':email', $email , PDO::PARAM_STR);
    $this->statement->execute();
}
```

Poi ho implementato un nuovo metodo nel MailModel

```
public function emailUpdatePassword($emailUser, $newPassword){  
    $this->mail->addAddress($emailUser);  
    $this->mail->AltBody = "<b>La tua password è stata aggiornata con successo<b>";  
    $this->mail->Body = "Il tua password è stato aggiornata con successo, adesso non  
dovrai fare altro che accedere al sito con la password di default: <br>"  
        . $newPassword .  
        "<br> e cambiare la password con una tua personale,<br> Cordiali saluti,<br>  
Mattia</p>";  
    return $this->sendEmail();  
}
```

Dato ripetevo sempre la stessa per inviare l'email ho aggiunto questo metodo:

```
public function sendEmail(){  
    if ($this->mail->send()) {  
        return true;  
    } else {  
        return false;  
    }  
}
```

E ho messo alcune variabili per inviare l'email nel costruttore.

Poi ho fatto in modo che l'utente loggato non può autoeliminarsi, ho bloccato i bottoni per utenti normali di aggiunta e rimozione delle vasche.

Ho sistemato altri piccoli particolari ed infine ho iniziato l'implementazione della gestione degli abitanti.

Problemi riscontrati e soluzioni adottate

-

Punto della situazione rispetto alla pianificazione

In anticipo di 4ore.

Programma di massima per la prossima giornata di lavoro

-Finire la gestione degli abitanti delle vasche