

# Diario di lavoro

Luogo	Trevano
Data	03.09.2019

## Lavori svolti

Oggi durante la prima ora abbiamo avuto la possibilità di leggere tutti i quaderni dei compiti di tutti e venti i progetti. Al termine dell'ora abbiamo potuto scegliere quale progetto avremmo voluto seguire ed abbiamo ricevuto il QdC che abbiamo iniziato a leggere. Al termine della pausa ho avuto la possibilità di chiedere delle spiegazioni riguardo alcuni punti che non erano molto chiari con il docente responsabile Luca Peduzzi. Fatto ciò ho iniziato a pensare alla base da utilizzare per il codice ed ho deciso di utilizzare il pattern MVC. Dopodiché mi sono informato sulle possibilità esistenti per creare le mappe e per trovare le coordinate di un posto tramite il suo indirizzo, ho trovato le API di Google Geocoding la scelta migliore perché molto precise anche se la versione di sviluppo (gratis) ha delle possibilità in meno. Dopodiché mi sono occupato di creare una repository per il progetto su GitHub, software che utilizzerò come gestione delle versioni, ed iniziato a creare le cartelle per la documentazione, diari, presentazione e codice. Nella cartella del codice chiamata application ho inserito la base MVC fornitaci dal docente Sartori l'anno scorso.

## Problemi riscontrati e soluzioni adottate

Oggi non ho riscontrato problemi.

## Punto della situazione rispetto alla pianificazione

Sono in orario.

## Programma di massima per la prossima giornata di lavoro

La prossima giornata voglio fare il diagramma di gantt ed iniziare l'analisi del dominio e se ho tempo l'analisi delle specifiche e requisiti.

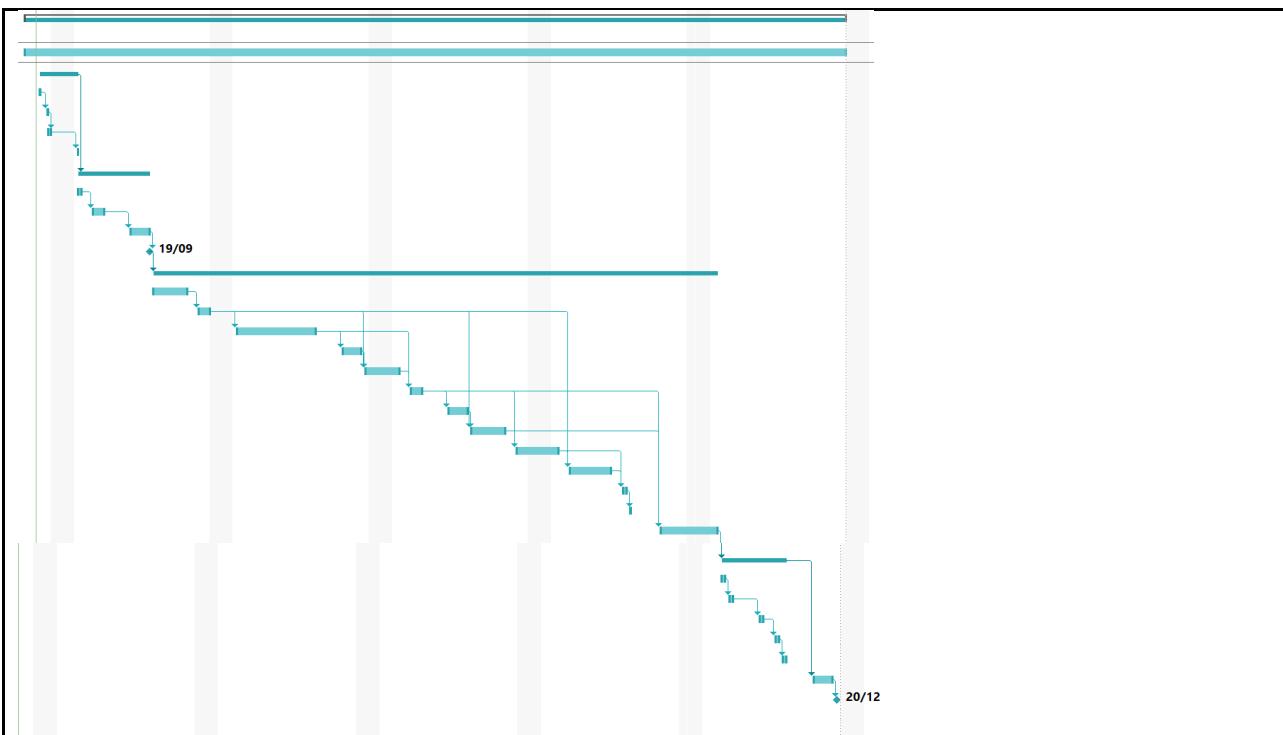
## Diario di lavoro

Luogo	Trevano
Data	05.09.2019

## Lavori svolti

Oggi per la prima parte della lezione ho installato il software per creare i diagrammi di Gantt Microsoft Project grazie ad un installer datoci da Valsangiacomo durante il secondo anno. Alla fine dell'installazione ho iniziato a creare il diagramma inserendo le attività che secondo me saranno le più importanti del progetto e assegnandogli una tempistica.

Piattaforma per la scoperta dei grotti del Cantone Ticino	Tue 03/09/19 13:15	Fri 20/12/19 16:30
Documentazione	Tue 03/09/19 13:15	Fri 20/12/19 16:30
<b>Analisi</b>		
Diagramma di Gantt	Thu 05/09/19 13:15	Thu 05/09/19 13:30
Analisi del dominio	Fri 06/09/19 13:15	Fri 06/09/19 15:30
Analisi dei requisiti	Fri 06/09/19 15:30	Fri 06/09/19 16:30
Analisi dei costi	Tue 10/09/19 13:15	Tue 10/09/19 14:30
<b>Progettazione</b>		<b>3</b>
Design dell'architettura del sistema	Tue 10/09/19 14:30	Tue 10/09/19 16:30
Design banca dati	Thu 12/09/19 13:15	Fri 13/09/19 16:30
Design delle interfacce	Tue 17/09/19 13:15	Thu 19/09/19 16:30
Fine progettazione	Thu 19/09/19 16:30	Thu 19/09/19 16:30
<b>Implementazione</b>		<b>12</b>
Creazione database	Fri 20/09/19 13:15	Tue 24/09/19 16:30
Scrittura codice di connessione al DB	Thu 26/09/19 13:15	Fri 27/09/19 16:30
Creazione pagina principale	Tue 01/10/19 13:15	Fri 11/10/19 16:30
Creazione pagina di registrazione	Tue 15/10/19 13:15	Thu 17/10/19 16:30
Scrittura codice di gestione della registrazione	Fri 18/10/19 13:15	Thu 22/10/19 16:30
Creazione pagina di login	Thu 24/10/19 13:15	Fri 25/10/19 16:30
Scrittura codice di controllo dei login	Tue 29/10/19 13:15	Thu 31/10/19 16:30
Scrittura codice di verifica se l'utente è admin	Fri 01/11/19 13:15	Tue 05/11/19 16:30
Creazione pagina di aggiunta grotto	Thu 07/11/19 13:15	Tue 12/11/19 16:30
Scrittura codice di gestione dell'indirizzo tramite API di Google Geocoding	Thu 14/11/19 13:15	Thu 19/11/19 16:30
Scrittura codice di controllo dell'inserimento	Thu 21/11/19 13:15	Thu 21/11/19 16:30
Creazione pagina di ricerca	Fri 22/11/19 13:15	Fri 22/11/19 14:15
Creazione pagina amministratore	Tue 26/11/19 13:15	Tue 03/12/19 16:30
Termino sviluppo	Tue 03/12/19 16:30	Tue 03/12/19 16:30
<b>Test</b>		<b>27</b>
Test funzionamento pagina principale	Thu 05/12/19 13:15	Thu 05/12/19 16:30
Test funzionamento registrazione e login	Fri 06/12/19 13:15	Fri 06/12/19 16:30
Test funzionamento pagina aggiunta grotto	Tue 10/12/19 13:15	Tue 10/12/19 16:30
Test funzionamento pagina admin	Thu 12/12/19 13:15	Thu 12/12/19 16:30
Test sicurezza generale del sistema	Fri 13/12/19 13:15	Fri 13/12/19 16:30
Presentazione	Tue 17/12/19 13:15	Thu 19/12/19 16:30
Fine progetto	Fri 20/12/19 13:15	Fri 20/12/19 13:15



Come si può vedere dalle immagini gran parte del tempo è stata stimata di venire utilizzata durante l'implementazione. Questo è dovuto alle innumerevoli pagine che dovrò implementare sia frontend che backend. La documentazione dura tutto il tempo dato che essa verrà portata avanti con l'avanzare del progetto. La pianificazione è stata fatta dando molto tempo allo sviluppo perché prevedo che ci saranno lezioni in cui riscontrerò problemi e necessiterò di tempo per risolverli.

Dopo aver terminato il diagramma di Gantt ho scritto i capitoli “Informazioni sul progetto” e “Scopo” della documentazione.

Non sono riuscito ad iniziare l’analisi del dominio come programmato martedì dato che il Gantt mi ha preso parecchio tempo.

#### Problemi riscontrati e soluzioni adottate

Oggi non ho riscontrato problemi tranne alcune difficoltà nel ricordarmi il funzionamento di determinate funzionalità di MS Project che ho risolto cercando sui forum ufficiali.

#### Punto della situazione rispetto alla pianificazione

Sono in orario.

#### Programma di massima per la prossima giornata di lavoro

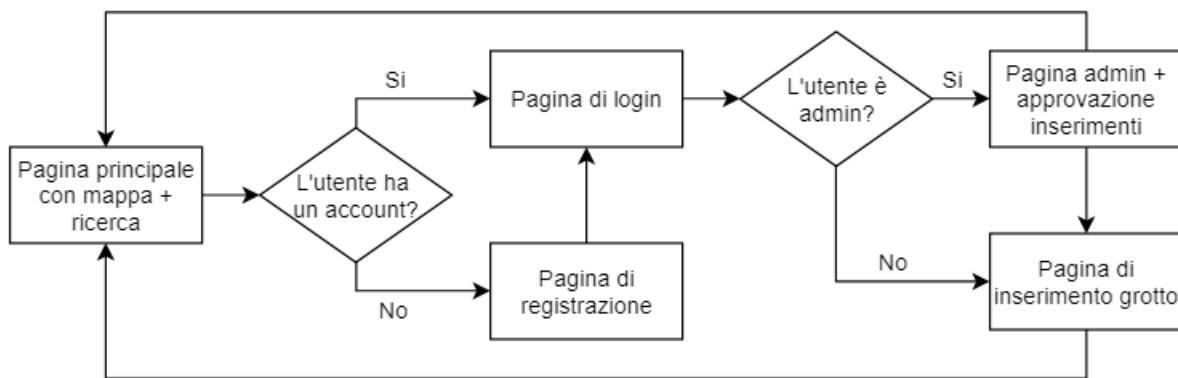
La prossima giornata voglio scrivere i capitoli della documentazione “analisi del dominio” e “analisi dei requisiti” e se avrò tempo iniziare il design dell’architettura del sistema.

# Diario di lavoro

Luogo	Trevano
Data	06.09.2019

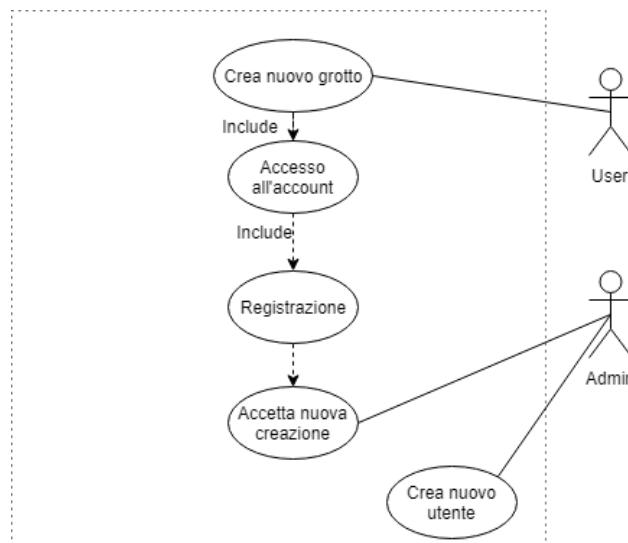
## Lavori svolti

Oggi ho fatto come da programma il capitolo della documentazione “analisi del dominio” ed una prima bozza che sarà da discutere con il docente del capitolo “analisi e specifica dei requisiti”. Ho anche avuto tempo per iniziare e completare il design dell’architettura del sistema che si presenta così:



Esso rappresenta il flusso che si seguirà in teoria per muoversi nel sito e parte con la pagina principale che consentirà di passare ad una pagina di login/registrazione dopodiché si passerà, se non si è admin, alla pagina di creazione di un grotto mentre, se si hanno i privilegi di amministratore, alla pagina riservata ad essi.

Completato il diagramma dell’architettura del sistema ho iniziato a fare quello degli use case ma non sono riuscito a terminarlo. Questo è quello che ho fatto per oggi:



**Problemi riscontrati e soluzioni adottate**

Oggi gli unici problemi che ho riscontrato erano nel esportare il diagramma di gantt in una forma leggibile dato che esso è piuttosto grande. Ho inoltre avuto alcune difficoltà nell'iniziare li use case dato che non gli avevo mai fatti.

**Punto della situazione rispetto alla pianificazione**

Sono in orario.

**Programma di massima per la prossima giornata di lavoro**

La prossima giornata voglio terminare gli ultimi capitoli di analisi quindi Use Case, spiegazione del gantt, hardware e software ed iniziare il diagramma ER dell'architettura del database.

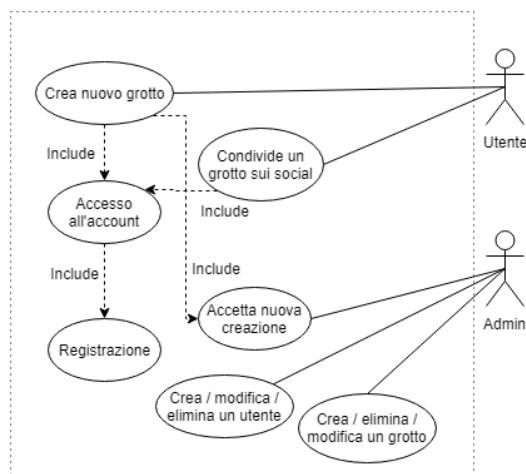
# Diario di lavoro

Luogo	Trevano
Data	10.09.2019

## Lavori svolti

La giornata di oggi è stata piuttosto incentrata sulla documentazione, ho infatti completato tutti i capitoli di analisi esclusa l'analisi dei mezzi che non posso ancora fare perché non conosco con certezza tutte le librerie che andrò ad utilizzare.

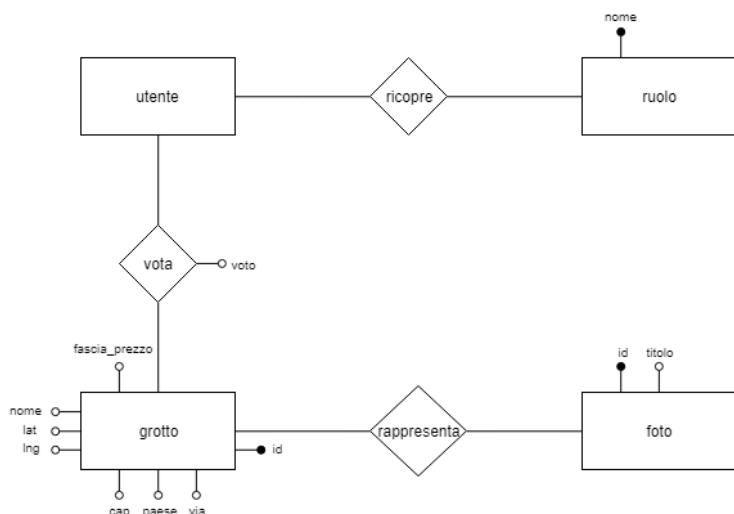
Oltre alla documentazione ho completato il diagramma degli use case



Esso ora include pure la condivisione sui social e tutte le attività dell'admin che non erano state inserite la volta scorsa.

Fatto ciò ho iniziato a redigere il diagramma ER del database prima su carta e poi su PC ma non ho avuto il tempo di terminare la copia sul computer.

Per adesso lo schema sul PC è il seguente:



Esso comprende quattro tabelle, le due più importanti sono utente e grotto che possiedono parecchi attributi (nello schema attuale non vi sono tutti). La tabella ruolo serve a specificare se un utente è admin o utente normale e l'ho fatta così che se in futuro si volesse aggiungere un nuovo ruolo (es. gerente grotto) si può fare senza problemi. Nella tabella grotto ci sono sia gli attributi latitudine e longitudine che l'indirizzo, questo è stato fatto perché contattare tutte le volte le API di Google Maps renderebbe il sito lento.

Problemi riscontrati e soluzioni adottate

Oggi non ho riscontrato problemi.

Punto della situazione rispetto alla pianificazione

Sono in orario.

Programma di massima per la prossima giornata di lavoro

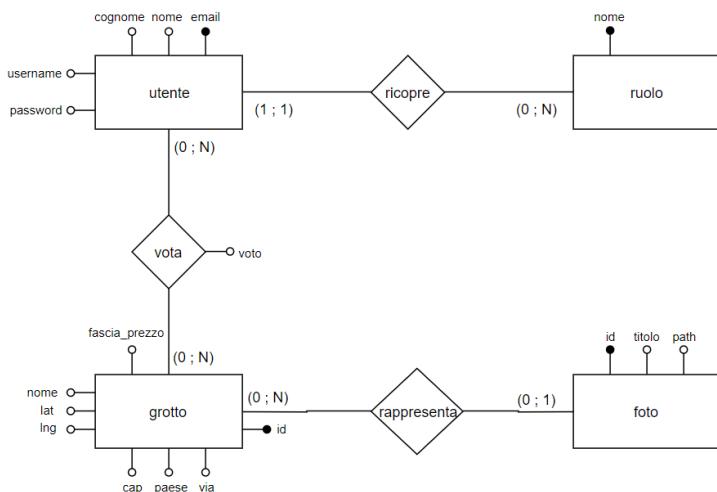
La prossima giornata voglio terminare l'ER e iniziare il design delle interfacce.

# Diario di lavoro

Luogo	Trevano
Data	12.09.2019

## Lavori svolti

Oggi ho completato il diagramma ER del database aggiungendo le part mancanti dall'ultima volta (gli attributi di alcune tabelle e le cardinalità) e l'ho spiegato sulla documentazione.



Mi sono reso conto durante la descrizione del capitolo nella documentazione che nella tabella immagini mancava un attributo "path" che ho poi aggiunto. Ho anche creato lo schema logico del DB.

```

RUOLO (nome)
UTENTE (email, nome, cognome, username, password, nome_ruolo(FK))
GROTO (id, via, paese, cap, lat, lng, nome, fascia_prezzo)
FOTO (id, titolo, path, id_grotto*(FK)) path unique
VOTO (email_utente(FK), id_grotto(fk), voto)
    
```

Fatto ciò sono tornato indietro nella documentazione al capitolo precedente e l'ho documentato, il diagramma del funzionamento del sistema era già stato fatto ma non era mai stato inserito nella documentazione.

Infine ho iniziato a creare il design delle interfacce, per adesso ho sviluppato quello dell'index, della pagina di login e di quella di registrazione.

Problemi riscontrati e soluzioni adottate

Oggi non ho riscontrato problemi.

Punto della situazione rispetto alla pianificazione

Sono leggermente in anticipo (metà lezione circa).

Programma di massima per la prossima giornata di lavoro

La prossima giornata voglio terminare il design delle interfacce e forse terminare la progettazione.

# Diario di lavoro

Luogo	Trevano
Data	13.09.2019

## Lavori svolti

Oggi ho completato i design delle interfacce creando quello della pagina di creazione di un grotto e quello della pagina admin. Inoltre l'ho documentato nell'apposito capitolo.

Grotti Ticino	Amministrazione	Home
Nome <input type="text"/> Fascia_Prezzo <input type="text"/> Valutazione <input type="text"/> CAP <input type="text"/> Paese <input type="text"/> Via <input type="text"/> No. Civico <input type="text"/>	La pagina di creazione di un grotto comprenderà i campi d'inserimento del nome, della fascia di prezzo, della valutazione e dell'indirizzo. Inoltre se si vorrà, tramite il tasto verifica, si potrà controllare se la posizione del puntatore viene messa nel punto giusto sulla mappa.	
<input type="button" value="Verifica"/> <input type="button" value="Salva"/> 		

Grotti Ticino	Aggiunta Grotto	Home																																																																							
Crea, modifica, elimina utenti <input type="button"/> Crea, modifica, elimina grotti <input type="button"/> Verifica nuovi inserimenti <input type="button"/>	La pagina admin di base contiene solamente tre buttoni. Premendo su il primo si aprirà un modale che consentirà di gestire gli utenti, con il secondo uno che farà gestire i grotti e con il terzo si potranno accettare/rifiutare gli inserimenti.																																																																								
	<table border="1"> <thead> <tr> <th>email</th> <th>username</th> <th>nome</th> <th>cognome</th> <th>modifica</th> <th>elimina</th> </tr> </thead> <tbody> <tr> <td>a@a.a</td> <td>a_b</td> <td>a</td> <td>b</td> <td>modifica</td> <td>elimina</td> </tr> <tr> <td>b@b.b</td> <td>c_d</td> <td>c</td> <td>d</td> <td>modifica</td> <td>elimina</td> </tr> <tr> <td>c@c.c</td> <td>e_f</td> <td>e</td> <td>f</td> <td>modifica</td> <td>elimina</td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th>nome</th> <th>indirizzo</th> <th>lat</th> <th>lon</th> <th>modifica</th> <th>elimina</th> </tr> </thead> <tbody> <tr> <td>a</td> <td>6900 a, a 10</td> <td>1</td> <td>2</td> <td>modifica</td> <td>elimina</td> </tr> <tr> <td>b</td> <td>6900 b, b 10</td> <td>3</td> <td>4</td> <td>modifica</td> <td>elimina</td> </tr> <tr> <td>c</td> <td>6900 c, c 10</td> <td>5</td> <td>6</td> <td>modifica</td> <td>elimina</td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th>user</th> <th>grotto</th> <th>voto</th> <th>prezzo</th> <th>accetta</th> <th>rifiuta</th> </tr> </thead> <tbody> <tr> <td>a</td> <td>6900 a, a 10</td> <td>1</td> <td>2</td> <td>accetta</td> <td>rifiuta</td> </tr> <tr> <td>b</td> <td>6900 b, b 10</td> <td>3</td> <td>4</td> <td>accetta</td> <td>rifiuta</td> </tr> <tr> <td>c</td> <td>6900 c, c 10</td> <td>5</td> <td>6</td> <td>accetta</td> <td>rifiuta</td> </tr> </tbody> </table>	email	username	nome	cognome	modifica	elimina	a@a.a	a_b	a	b	modifica	elimina	b@b.b	c_d	c	d	modifica	elimina	c@c.c	e_f	e	f	modifica	elimina	nome	indirizzo	lat	lon	modifica	elimina	a	6900 a, a 10	1	2	modifica	elimina	b	6900 b, b 10	3	4	modifica	elimina	c	6900 c, c 10	5	6	modifica	elimina	user	grotto	voto	prezzo	accetta	rifiuta	a	6900 a, a 10	1	2	accetta	rifiuta	b	6900 b, b 10	3	4	accetta	rifiuta	c	6900 c, c 10	5	6	accetta	rifiuta
email	username	nome	cognome	modifica	elimina																																																																				
a@a.a	a_b	a	b	modifica	elimina																																																																				
b@b.b	c_d	c	d	modifica	elimina																																																																				
c@c.c	e_f	e	f	modifica	elimina																																																																				
nome	indirizzo	lat	lon	modifica	elimina																																																																				
a	6900 a, a 10	1	2	modifica	elimina																																																																				
b	6900 b, b 10	3	4	modifica	elimina																																																																				
c	6900 c, c 10	5	6	modifica	elimina																																																																				
user	grotto	voto	prezzo	accetta	rifiuta																																																																				
a	6900 a, a 10	1	2	accetta	rifiuta																																																																				
b	6900 b, b 10	3	4	accetta	rifiuta																																																																				
c	6900 c, c 10	5	6	accetta	rifiuta																																																																				

Durante la lezione mi sono reso conto di aver dimenticato due attributi nella tabella grotto e quindi gli ho aggiunto i campi **no\_civico** e **valutazione**.  
 Fatto ciò ho creato il database su sql con il seguente codice:

Piattaforma per la scoperta dei Grotti del Canton Ticino

```
create database grotti;
use grotti;

create table ruolo(
    nome varchar(50) primary key
);

create table utente(
    email varchar(50) primary key,
    username varchar(50) not null,
    nome varchar(50) not null,
    cognome varchar(50) not null,
    password varchar(255) not null,
    nome_ruolo varchar(50) not null,
    foreign key(nome_ruolo) references ruolo(nome)
);

create table grotto(
    id int primary key auto_increment,
    nome varchar(50) not null,
    lon double not null,
    lat double not null,
    no_civico varchar(10) not null,
    via varchar(50) not null,
    paese varchar(50) not null,
    cap int not null,
    fascia_prezzo enum("Buon mercato", "Nella norma", "Caro") not null,
    valutazione int not null
);

create table foto(
    id int primary key auto_increment,
    titolo varchar(50) not null,
    path varchar(50) not null
);

create table voto(
    email_utente varchar(50),
    id_grotto int,
    voto int not null,
    primary key(email_utente, id_grotto),
    foreign key(email_utente) references utente(email),
    foreign key(id_grotto) references grotto(id)
);
```

**Problemi riscontrati e soluzioni adottate**  
Oggi non ho riscontrato problemi.

**Punto della situazione rispetto alla pianificazione**  
Sono in anticipo (circa una lezione).

**Programma di massima per la prossima giornata di lavoro**  
La prossima giornata voglio creare la classe di collegamento al db e iniziare a sviluppare l'header e il footer. Inoltre dopo aver discusso con Paolo Weissaupt mi sono reso conto che sarebbe meglio creare una tabella a parte per la fascia di prezzo e anche questo lo farò la prossima volta.



# Diario di lavoro

Luogo	Trevano
Data	19.09.2019

## Lavori svolti

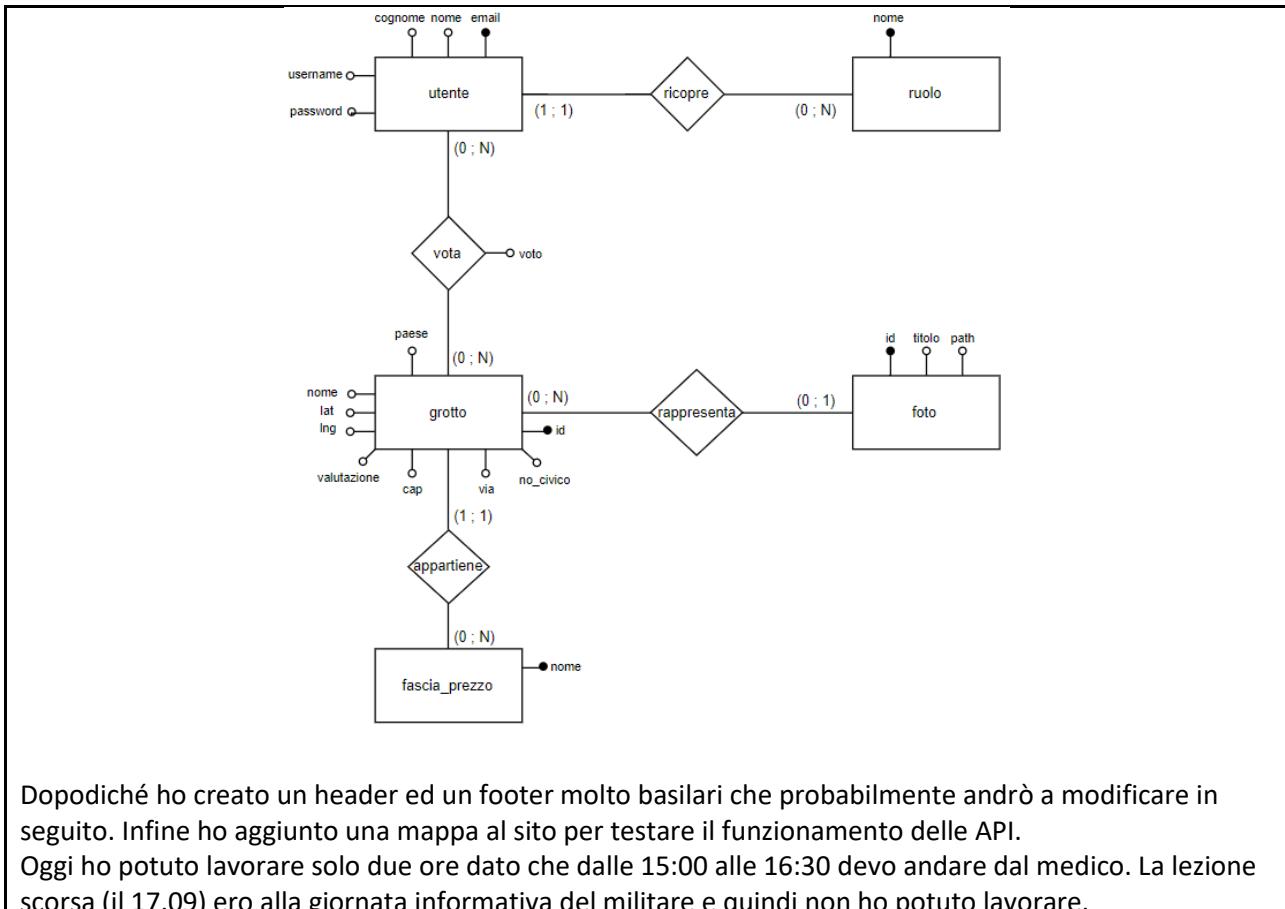
Oggi ho creato la classe di collegamento al database in PHP utilizzando PDO.

```
class db_connection
{
    private $_connection;

    private function __construct()
    {
    }

    /**
     * @return PDO La connessione con il database
     */
    public function getConnection()
    {
        if($this->_connection == null){
            try{
                $this->_connection = new PDO(DSN, USER, PASSWORD);
                return $this->_connection;
            }catch(PDOException $e){
                print "Error!: " . $e->getMessage() . "<br/>";
                die();
            }
        }
    }
}
```

Fatto ciò ho messo a posto l'ER del database e di conseguenza lo schema logico e il DB su mysql. Adesso, dopo aver aggiunto la tabella fascia\_prezzo, l'ER appare così.



Dopodiché ho creato un header ed un footer molto basilari che probabilmente andrò a modificare in seguito. Infine ho aggiunto una mappa al sito per testare il funzionamento delle API.  
Oggi ho potuto lavorare solo due ore dato che dalle 15:00 alle 16:30 devo andare dal medico. La lezione scorsa (il 17.09) ero alla giornata informativa del militare e quindi non ho potuto lavorare.

#### Problemi riscontrati e soluzioni adottate

Oggi non ho riscontrato problemi.

#### Punto della situazione rispetto alla pianificazione

Sono in anticipo (circa mezza lezione).

#### Programma di massima per la prossima giornata di lavoro

La prossima giornata voglio iniziare a implementare la pagina principale.

# Diario di lavoro

Luogo	Trevano
Data	20.09.2019

## Lavori svolti

Oggi ho iniziato l'implementazione della pagina iniziale. Come prima cosa ho messo la mappa in maniera responsive al centro della pagina. Dopodiché ho caricato, tramite la classe di connessione al database, tutti i grotti così da iniziare a stampare la lista di essi. Fatto ciò ho implementato, grazie al componente di material design bootstrap chiamato DataTable, i filtri su tutte le colonne. La documentazione con gli esempi si può trovare a questo link: <https://mdbootstrap.com/docs/jquery/tables/datatables/>. Infine ho aggiunto la colonna "telefono" alla tabella grotto di cui mi ero dimenticato ed ho cambiato il tipo di dato della colonna "valutazione" da int a double.

## Problemi riscontrati e soluzioni adottate

Oggi non ho riscontrato problemi.

## Punto della situazione rispetto alla pianificazione

Sono in anticipo (circa mezza lezione).

## Programma di massima per la prossima giornata di lavoro

La prossima giornata voglio terminare l'implementazione della pagina principale e iniziare la pagina di registrazione.

# Diario di lavoro

Luogo	Trevano
Data	24.09.2019

## Lavori svolti

Oggi ho continuato con l'implementazione della pagina principale. Inizialmente ho fatto sì che la mappa caricasse i marker dal database, questo è fatto grazie ad una funzione da me scritta che per ogni grotto nel DB genera un marker.

Fatto ciò ho aggiunto alla stessa funzione il codice che crea la finestra contenente i dettagli di un grotto che viene visualizzata quando si clicca su di un marker. Essa mi ha impiegato parecchio tempo dato che ho avuto dei problemi con JavaScript e PHP.

Inizialmente l'idea era quella di caricare i marker con PHP e passarli a javascript con cui li avrei creati ma dopo un po' di tentativi ho capito che questa non era la scelta migliore, ho quindi deciso di utilizzare l'array di PHP per ciclare i grotti e inserire i valori. Tutto ciò viene fatto nella funzione setMarkers().

Essa inizia con un loop di PHP:

```
function setMarkers(map, locations) {
    <?php foreach ($_SESSION['grotti'] as $row): ?>
    ...
}
```

Dopodiché contiene una variabile che rappresenta il codice html da creare che viene passata all'infowindow, questo è l'oggetto delle API di Google fatto apposta per creare delle finestre sopra i marker.

```
var contentString = `
<div class='content modal-body'>
    <h1 id="nome" style='color:black;'><?php echo $row['nome']; ?> </h1>
    <strong id="indirizzo">Indirizzo</strong>
    <?php echo(" ". $row['cap'] . ". " . $row['paese'] . ", "
    $row['via'] . ". " . $row['no_civico']); ?>
    <br>
    <strong id="telefono">Telefono</strong><?php echo " " .
    $row['telefono']; ?>
    <br>
    <strong id="valutazione">Valutazione</strong><?php echo " " .
    $row['valutazione']; ?>
    <br><br>
</div>`;

var infowindow<?php echo $row['id']; ?> = new google.maps.InfoWindow({
    content: contentString
});
```

Infine crea il marker e ad ogni marker associa un listener che, al click sul punto della mappa, genera l'infowindow e ci zooma sopra.

```
var marker<?php echo $row['id']; ?> = new google.maps.Marker({
    position: luogo,
    map: map,
    animation: google.maps.Animation.DROP,
    title: <?php echo "" . $row['nome'] . ""; ?>
});
```

```
marker<?php echo $row['id']; ?>.addListener('click', function() {  
    if (infowindowOpen != null) {  
        infowindowOpen.close();  
    }  
    infowindow<?php echo $row['id']; ?>.open(map, marker<?php echo  
$row['id']; ?>);  
    infowindowOpen = infowindow<?php echo $row['id']; ?>;  
  
    map.setZoom(13);  
    map.setCenter(marker<?php echo $row['id']; ?>.getPosition());  
});
```

#### Problemi riscontrati e soluzioni adottate

Oggi ho inizialmente riscontrato problemi con il passaggio dell'array da PHP a Javascript, utilizzando il metodo descritto da tutte le guide l'array era sempre nullo e non capivo il perché. Dopo aver chiesto aiuto a Filippo Finke e aver cercato l'errore per circa 15 minuti siamo giunti alla conclusione che si trattava di un problema di encoding, ho quindi provato a cambiare il charset di mysql da UTF-8 a UTF-8mb4 ma anche questo non ha portato a grandi risultati. Dopo ulteriori ricerche ho capito che era PHP che quando si connetteva al DB prendeva i dati giusti ma avendo lui l'encoding su UTF-8 li cambiava. Ho quindi risolto il problema aggiungendo il codice seguente al DSN.

```
define ('DSN', 'mysql:host=127.0.0.1;dbname=grotti; charset=utf8mb4');
```

Il secondo problema riscontrato è stato riguardo l'inserimento dei marker sulla mappa, infatti avevo provato in un modo presentato in una guida ma non ha funzionato. Esso creava infatti i marker ma avevano tutti lo stesso titolo e infowindow. Dopo vari tentativi ho ripreso un sito che abbiamo fatto io e Luca Di Bello durante il nostro stage in Irlanda e l'ho modificato. La soluzione finale è quella rappresentata sopra nel punto «Lavori svolti».

#### Punto della situazione rispetto alla pianificazione

Sono in orario.

#### Programma di massima per la prossima giornata di lavoro

La prossima giornata voglio terminare l'implementazione della pagina principale e iniziare la pagina di registrazione.

# Diario di lavoro

Luogo	Trevano
Data	26.09.2019

## Lavori svolti

Oggi ho continuato con l'implementazione della pagina principale. Inizialmente ho messo a posto un piccolo problema che ho notato nella pagina principale: lo zoom sulla mappa non sempre eseguiva l'animazione e l'ho risolto cambiando il valore che esso doveva raggiungere (io richiedevo uno zoom di 7 quando il minimo era 9). Dopodiché ho iniziato a implementare il frontend della pagina di login, questo mi serviva dato che è da essa che si può passare a quella di registrazione. Grazie al framework, Material Design Bootstrap, che sto utilizzando l'operazione è stata piuttosto veloce. Fatto ciò ho creato anche il frontend della pagina di registrazione con lo stesso metodo e stile di quella di login. Ho poi continuato facendo i controlli backend dei dati inseriti, essi verificano inizialmente che il metodo di connessione sia corretto (post), poi che tutti i campi siano stati compilati ed infine richiama una funzione che elimina da essi possibili caratteri speciali e malevoli.

La classe è la seguente:

```
class InputManager {
    public function checkInput($input) {
        $input = trim($input);
        $input = stripslashes($input);
        $input = htmlspecialchars($input);
        return $input;
    }
}
```

Una volta controllati tutti i valori essi vengono controllati con degli if che ne verificano la lunghezza.

```
$im = new InputManager();
$firstname = filter_var($im->checkInput($_POST['firstname']), FILTER_SANITIZE_STRING);
$lastname = filter_var($im->checkInput($_POST['lastname']), FILTER_SANITIZE_STRING);
$username = filter_var($im->checkInput($_POST['username']), FILTER_SANITIZE_STRING);
$email = filter_var($im->checkInput($_POST['email']), FILTER_SANITIZE_EMAIL);
$password = filter_var($im->checkInput($_POST['password']), FILTER_SANITIZE_STRING);
$repassword = filter_var($im->checkInput($_POST['repassword']), FILTER_SANITIZE_STRING);

if(!(strlen($firstname) > 0 && strlen($firstname) < 50)){
    array_push($errors, "Il nome deve essere lungo tra gli 1 e 50 caratteri");
}
if(!(strlen($lastname) > 0 && strlen($lastname) < 50)){
    array_push($errors, "Il cognome deve essere lungo tra gli 1 e 50 caratteri");
}
if(!(strlen($username) > 0 && strlen($username) < 50)){
    array_push($errors, "Lo username deve essere lungo tra gli 1 e 50 caratteri");
}
if(!(strlen($email) > 0 && strlen($email) < 50)){
    array_push($errors, "L'email deve essere formattata nel seguente modo:
    indirizzo@dominio.xx");
}
if(!(strlen($password) >= 8)){
    array_push($errors, "La password deve essere almeno lunga 8 caratteri");
}
```

Se anche qui tutto risulta corretto si passa a controllare che le due password siano uguali e che l'email non sia già in uso.

```
if($password == $repassword) {  
    $db = (new db_connection)->getUsers();  
  
    foreach ($db as $row) {  
        if ($row['email'] == $email) {  
            array_push($errors, "L'email è già in uso");  
            $exists = true;  
        }  
    }  
}
```

Se tutto risulta corretto si procede con l'inserire l'utente nel database, questo viene fatto richiamando la seguente funzione.

```
public function addUser($firstname, $lastname, $username, $email, $password) {  
    $db = $this->getConnection();  
    $query = $db->prepare('insert into utente(email, nome, cognome, username,  
password, `nome_ruolo`) values (:email, :firstname, :lastname, :username,  
:password, :nome_ruolo')');  
  
    $password = hash('sha256', $password);  
    $type = 'utente';  
  
    $query->bindParam(':email', $email, PDO::PARAM_STR);  
    $query->bindParam(':firstname', $firstname, PDO::PARAM_STR);  
    $query->bindParam(':lastname', $lastname, PDO::PARAM_STR);  
    $query->bindParam(':username', $username, PDO::PARAM_STR);  
    $query->bindParam(':password', $password, PDO::PARAM_STR);  
    $query->bindParam(':nome_ruolo', $type, PDO::PARAM_STR);  
  
    $query->execute();  
}
```

Essa esegue l'insert nella tabella utilizzando i prepared statements che impediscono le SQL Injections.

Se nelle verifiche si riscontrano errori si dovrà rimandare l'utente alla pagina di login e mostrare i messaggi d'errore.

#### Problemi riscontrati e soluzioni adottate

Oggi ho avuto un solo grande problema che è stato il fatto che, dopo aver inserito l'utente nel database, richiamando la pagina di login lo stile non venisse caricato. Questo era dovuto al fatto che il require (che utilizzavo io) non ricarica la pagina ma aggiunge alla corrente il contenuto di un file php e quindi non si caricavano gli stili. Ho risolto il problema chiedendo aiuto al docente Sartori che mi ha consigliato di utilizzare il metodo header al posto di require.

#### Punto della situazione rispetto alla pianificazione

Sono in orario.

#### Programma di massima per la prossima giornata di lavoro

La prossima giornata voglio terminare l'implementazione della pagina di registrazione e iniziare la pagina di login (backend).

# Diario di lavoro

Luogo	Trevano
Data	27.09.2019

## Lavori svolti

Oggi ho continuato con l'implementazione della pagina di registrazione, non sono riuscito a fare molto perché subito dall'inizio mi sono bloccato con un problema con la scrittura nel database. Risolto il problema, dopo le prime due ore, sono dovuto andare ad una riunione di resoconto finale dello stage all'estero che ho fatto quest'estate e quindi le due ore seguenti non ho avuto l'occasione di fare molto. La prossima volta testerò tutta la pagina ma in teoria è finita.

## Problemi riscontrati e soluzioni adottate

Dopo un'ora di lezione a provare a capire dove avesse problemi il metodo di inserimento di un utente nel database sono riuscito a trovarne il problema. Questo era nel metodo che scrive nel db che, utilizzando i prepared statements, non scriveva ma non ritornava nemmeno errori, cercando su internet ho trovato un secondo metodo di utilizzare i prepared statements e utilizzandolo il problema non si presenta più. Il metodo di scrittura nel db addUser() ora si presenta così:

```
public function addUser($firstname, $lastname, $username, $email, $password) {
    try {
        $db = $this->getConnection();
        $query = $db->prepare('INSERT INTO utente(email, nome, cognome, username,
password, nome_ruolo) VALUES (?, ?, ?, ?, ?, ?)');
        $password = hash('sha256', $password);
        $type = "utente";

        $query->bindParam(1, $email, PDO::PARAM_STR);
        $query->bindParam(2, $firstname, PDO::PARAM_STR);
        $query->bindParam(3, $lastname, PDO::PARAM_STR);
        $query->bindParam(4, $username, PDO::PARAM_STR);
        $query->bindParam(5, $password, PDO::PARAM_STR);
        $query->bindParam(6, $type, PDO::PARAM_STR);

        $query->execute();
    } catch (Exception $e) {
        header('Location: ' . URL . 'db_error');
    }
}
```

La parte finale, nel catch, serve a richiamare una pagina d'errore così da non mostrare all'utente finale un errore di PHP poco chiaro.

## Punto della situazione rispetto alla pianificazione

Sono in orario.

## Programma di massima per la prossima giornata di lavoro

La prossima giornata voglio iniziare la pagina di login (backend).

# Diario di lavoro

---

Luogo	Trevano
Data	01.10.2019

## Lavori svolti

Oggi ho iniziato con l'implementazione del backend della pagina di registrazione e l'ho portato a termine. Ho iniziato creando una funzione che verifica il login di un utente verificando che l'email sia in uso da un account e, se così fosse, che la password corrisponda con quella dell'account.

```
public function checkLogin(){
    //richiamo le classi di cui avrò bisogno
    unset($_SESSION['warning']);
    require_once "./application/models/db_connection.php";
    require_once "./application/models/input_manager.php";

    $errors = array();

    //verifico il metodo di richiesta
    if($_SERVER["REQUEST_METHOD"] == "POST") {
        //verifico che i campi siano impostati e che non siano stringhe vuote
        if (isset($_POST['email']) && !empty($_POST['email']) &&
            isset($_POST['password']) && !empty($_POST['password'])) {

            //genero un nuovo InputManager e testo gli inserimenti
            $im = new InputManager();

            $email = filter_var($im->checkInput($_POST['email']),
FILTER_SANITIZE_EMAIL);
            $password = filter_var($im->checkInput($_POST['password']),
FILTER_SANITIZE_STRING);

            //prendo tutti gli utenti nel database
            $db = (new db_connection)->getUsers();

            //hash della password così da poterla comparare con quella nel db
            $password = hash('sha256', $password);
            foreach ($db->fetchAll() as $row) {
                //controllo che l'email sia in uso da un utente
                if ($row['email'] == $email) {
                    //controllo che la password corrisponda
                    if($row['password'] == $password){
                        $_SESSION['user'] = (new db_connection)->getUser($email);
                        //verifico se è admin o utente normale
                        if($row['nome_ruolo'] == 'admin'){
                            header('Location: ' . URL . 'admin');
                            exit();
                        }elseif($row['nome_ruolo'] == 'utente'){
                            header('Location: ' . URL . 'home');
                            exit();
                        }
                    }else{
                        //se la password è sbagliata ritorno l'errore
                        array_push($errors, "Password o email sbagliate");
                        $_SESSION['warning'] = $errors;
                        header('Location: ' . URL . 'login');
                        exit();
                    }
                }
            }
            //se l'email non è in utilizzo da nessun user ritorno l'errore
            array_push($errors, "Password o email sbagliate");
            $_SESSION['warning'] = $errors;
            header('Location: ' . URL . 'login');
            exit();
        }
    }
}
```

Dopo aver scritto e testato la funzione ho implementato alcune piccole cose nei controller ad esempio nell'home se si è loggati e a dipendenza di che livello di privilegi si hanno si vedono degli header diversi. Headers che sono andato a modificare oggi aggiungendo i percorsi alle pagine e inserendo nei due per gli utenti loggati la possibilità di uscire dall'account dato che ho creato la funzione.

```
public function logout(){
    unset($_SESSION['user']);
    header('Location: ' . URL . 'home');
    exit();
}
```

Ho inoltre commentato tutti i controllers e i models che non avevano ancora tutte le spiegazioni.

Per far sì che il login funzionasse ho implementato la funzione getUser nel model di gestione del database. Essa ritorna un utente in base alla sua email.

```
public function getUser($email){
    try{
        $db = $this->getConnection();

        $query = $db->prepare('SELECT * FROM utente where email=?');
        $query->bindParam(1, $email, PDO::PARAM_STR);

        $query->execute();

        return $query->fetchAll();
    }catch (Exception $e){
        header('Location: ' . URL . 'warning');
    }
}
```

#### Problemi riscontrati e soluzioni adottate

Inizialmente ho riscontrato un problema con il login dato che nel database avevo degli hash non corretti e quindi facendo il paragone la password risultava sempre sbagliata. Ho risolto eliminando i record e reinserendoli nel database dalla pagina di registrazione.

#### Punto della situazione rispetto alla pianificazione

Sono in anticipo sulla pianificazione.

#### Programma di massima per la prossima giornata di lavoro

La prossima giornata voglio creare i validatori frontend per le pagine di registrazione e di login.

# Diario di lavoro

Luogo	Trevano
Data	03.10.2019

## Lavori svolti

Oggi come da programma ho impostato i controlli frontend con javascript, essi vengono fatti con delle classi mie e di Luca Di Bello che utilizzavamo nello stage in Irlanda, di cui avevo già testato il funzionamento e sapevo come utilizzarle. Esse, al caricamento della pagina, assegnano a tutti gli input un listener sia on key up (bottone si alza) che on key down (pressione del bottone). Quando il listener viene richiamato nella classe register la funzione manage verifica quale è il campo utilizzato e richiama il validatore corretto dalla classe validate. Se la validazione va a buon fine la parte bassa dell'input si colora di verde altrimenti di rosso e appare un messaggio d'errore.

Fatto ciò ho creato una pagina interattiva che consente di, dopo aver premuto su una linea della tabella nella home, di visualizzare i dettagli e le immagini di un grotto. Per fare ciò ho dovuto creare una funzione che prende i dati dal database di un grotto in base all'id passato.

```
public function getGrotto($id) {
    try{
        $db = $this->getConnection();
        $query = $db->prepare('SELECT * FROM grotto WHERE id=?');
        $query->bindParam(1, $id);
        $query->execute();
        return $query->fetchAll();
    }catch (Exception $e){
        header('Location: ' . URL . 'warning');
    }
}
```

Ho anche scritto la funzione che ritorna tutte le immagini di un grotto così da mostrarle nella pagina sotto forma di un carousel (slide show).

```
public function getImages($id) {
    try{
        $db = $this->getConnection();
        $query = $db->prepare('SELECT * FROM foto WHERE grotto=?');
        $query->bindParam(1, $id);
        $query->execute();
        return $query->fetchAll();
    }catch (Exception $e){
        header('Location: ' . URL . 'warning');
    }
}
```

Inoltre ho aggiunto al database, nella tabella utente, i campi reset\_token (il token per resettare la password) e first\_login (un booleano che indica se l'utente è stato creato dall'admin o no).

## Problemi riscontrati e soluzioni adottate

L'unico problema che ho riscontrato oggi è che avevo dimenticato la foreign key del grotto nella tabella foto e in un secondo momento che la colonna path consentiva di inserire solamente dei campi di 50 caratteri ma un percorso è facilmente più lungo.

Ho risolto inserendo la colonna grotto e impostando path come varchar(255).

Punto della situazione rispetto alla pianificazione

Sono in anticipo sulla pianificazione.

Programma di massima per la prossima giornata di lavoro

La prossima giornata voglio completare la pagina di gestione degli errori e iniziare la pagina di aggiunta di un grotto.

# Diario di lavoro

Luogo	Trevano
Data	04.10.2019

## Lavori svolti

Oggi come da programma ho messo a posto la pagina d'errore, essa viene richiamata ad ogni errore con il DB e le viene passato il codice ed il messaggio d'errore. Nella pagina viene stampato un messaggio generico per gli utenti di base (C'è stato un errore! Aspetta o riprova ...) e anche il messaggio d'errore con il codice di MySQL per gli utenti che necessitano di risolvere il problema.

Fatto ciò ho iniziato a creare la pagina di aggiunta dei grotti, inizialmente ho creato la pagina con il form e impostato tutti i campi di esso dopodiché ho perso parecchio tempo cercando una libreria che consentisse di inserire nella pagina una valutazione fatta con le stelle. Dopo un po' di ricerche ho trovato «auxiliary-review» fatta appunto da «auxiliary». Ho impiegato circa mezz'ora a capire come farla funzionare dopodiché l'ho implementata.

Dato che ho perso parecchio tempo ad informarmi sulla libreria di valutazione non sono riuscito a procedere ulteriormente con la pagina.

Inoltre ho aggiunto al DB il campo verificato alla tabella grotto, esso serve per capire se un grotto è già stato verificato da un admin o meno. Se il grotto non è verificato apparirà nella pagina di amministrazione per essere verificato/scartato altrimenti mente se è già stato approvato verrà mostrato nella pagina principale.

## Problemi riscontrati e soluzioni adottate

Mi ha preso abbastanza tempo riuscire a far sì che il valore scelto dall'utente venga passato con il metodo post insieme agli altri elementi del form. Ho risolto il problema inserendo nel form un input nascosto a cui, tramite jQuery, assegno come valore il numero scelto dall'utente sotto forma di stelle.

```
//Ad ogni click setto come value dell'input nascosto val il valore della valutazione
$(".rating").click(function () {
    var value = String($(".rating").rate("getValue"));
    $('#val').val(value);
});
```

Un altro piccolo problema l'ho riscontrato cercando di centrare le stelle, facendolo tramite CSS infatti esse smettono di funzionare. Ho dovuto quindi farlo tramite JQuery.

```
//Mantengo sempre centrate le stelle
var margin = $('.rating-container').width()/2 - $('.rating').width()/2;
$('.rating').css('margin-left', margin);
$(window).resize(function () {
    var margin = $('.rating-container').width()/2 - $('.rating').width()/2;
    $('.rating').css('margin-left', margin);
});
```

Questo viene fatto una volta per quando viene caricata la pagina e poi ogni volta che la dimensione della finestra varia.

Punto della situazione rispetto alla pianificazione

Sono in anticipo sulla pianificazione.

Programma di massima per la prossima giornata di lavoro

La prossima giornata voglio completare la pagina di aggiunta di un grotto.

# Diario di lavoro

Luogo	Trevano
Data	08.10.2019

## Lavori svolti

Oggi come da programma ho continuato con la pagina di aggiunta di un grotto. Teoricamente essa è finita ma non sono in grado di testarlo dal momento che non funzionano le mappe di Google.

Per completarla ho dovuto mettere a posto la validazione front end e creare tutto il backend. Esso è piuttosto simile a quello della pagina di registrazione, inizialmente si verifica il metodo di invio del form dopodiché se tutti i campi sono stati compilati, se non vi sono problemi si passa a controllare che i campi non contengano stringhe malevoli e che la lunghezza di esse sia corretta.

Fatto ciò tramite la funzione addGrotto si aggiungerà un grotto al database, se sarà stato creato da un admin sarà già «verificato» altrimenti dovrà essere controllato da un admin.

Inoltre ho aggiunto alla pagina una mappa che consentirà, tramite il Geocoding, di verificare che l'indirizzo sia stato inserito correttamente dato che nel database verranno salvate le coordinate ritornate proprio dalle API di Geocoding.

## Problemi riscontrati e soluzioni adottate

Il problema che ho riscontrato oggi è che Google ha cambiato le politiche di fatturazione e ora richiedono una carta di credito per poter usare le loro API. Sfortunatamente la mia carta di credito è di Postfinance e loro non consentono a Google di effettuare prelievi automatizzati dalle carte di credito. Per questo motivo sto creando una carta virtuale di Revolut che teoricamente dovrebbe funzionare. Adesso sono in attesa di esser verificato.

## Punto della situazione rispetto alla pianificazione

Sono in anticipo sulla pianificazione.

## Programma di massima per la prossima giornata di lavoro

La prossima giornata voglio completare la pagina di aggiunta di un grotto e iniziare ad implementare i voti.

# Diario di lavoro

Luogo	Trevano
Data	10.10.2019

## Lavori svolti

Oggi come da programma ho provato a risolvere i problemi relativi alla pagina di creazione del grotto, di base la pagina funziona (inserisce nel DB) ma devo passare delle coordinate a caso hardcoded dato che la API di Google non funzionano e quindi non possono generare i dati dall'indirizzo.

Fatto ciò ho sviluppato il voto dei grotti, inizialmente ho fatto alcune modifiche al database dato che ho notato che aveva dei problemi, ho infatti modificato il tipo di dato del voto da int a float ed ho rimosso, nella tabella grotto, al campo votazione il «not null» così che adesso di base un grotto può non avere voti. Questo ha implicato la rimozione della votazione iniziale quando si crea un grotto dato che non mi sembrava corretto poterlo fare.

Per far funzionare il tutto ho creato un trigger in mysql, esso parte dopo ogni inserimento nella tabella voto ed esegue la media dei voti del grotto che ne ha appena ricevuto uno ed inserisce nella tabella grotto la nuova media.

```
/*CREAZIONE TRIGGER AFTER INSERT SULLA TABELLA VOTO*/
delimiter $$$
create trigger avg_after_review after insert on voto for each row
begin
    set @avg_review = (select avg(voto) from voto where id_grotto=NEW.id_grotto);
    update grotto set valutazione = @avg_review where id=NEW.id_grotto;
end;
$$$
delimiter ;
```

Fatto ciò ho aggiunto la valutazione con le stelle alla tabella iniziale e nei marker della mappa, quella nella tabella so che funziona ma purtroppo quella nella mappa non posso visualizzarla quindi non posso garantire il corretto funzionamento.

## Problemi riscontrati e soluzioni adottate

Oggi i problemi che ho riscontrato sono stati inizialmente relativi al trigger in mysql, infatti io lo avevo fatto seguendo la documentazione ufficiale ma non riuscivo a farlo funzionare. Dopo svariato tempo perso ho provato a chiedere al docente Sartori ma nemmeno lui è stato di aiuto. Alla fine sono riuscito a risolvere il problema chiedendo aiuto al docente Muggiasca che insegna un modulo su mysql che comprende i triggere e lui mi ha consigliato di aggiungere il delimiter.

Un altro problema che ho avuto è stato che non riuscivo a passare correttamente dalla pagina di votazione al database le votazioni con numeri non interi. Dopo un po' di ricerche ho risolto inserendo un'opzione al metodo di verifica dei numeri di PHP.

```
$valutazione=filter_var($im->checkInput($_POST['val']),FILTER_SANITIZE_NUMBER_FLOAT,
FILTER_FLAG_ALLOW_FRACTION);
```

L'opzione FILTER\_FLAG\_ALLOW\_FRACTION consente appunto di inserire numeri con la virgola e quindi il mio problema si è risolto.

## Punto della situazione rispetto alla pianificazione

Sono in anticipo sulla pianificazione.

Piattaforma per la scoperta dei Grotti del Canton Ticino

Programma di massima per la prossima giornata di lavoro

La prossima giornata voglio iniziare ad implementare il cambio della password e magari risolvere i problemi con le API di Google.

# Diario di lavoro

Luogo	Trevano
Data	11.10.2019

## Lavori svolti

Oggi durante le prime due ore non ho potuto lavorare dato che il docente Valsangiacomo ci ha spiegato come funzionano le valutazioni dei progetti di esame.

Nelle due ore dopo la pausa ho iniziato a scrivere il codice di reset della password, esso è suddiviso su un controller ed un model. Nel controller, dopo aver caricato un form con un input solo e aver atteso l'inserimento di un'email da parte dell'utente, verifico che i dati passati siano corretti e che l'email sia effettivamente in uso. Se tutti i requisiti sono soddisfatti viene richiamato un metodo nel model di gestione del database che genera un token e lo inserisce nel campo dell'utente «reset\_token».

```
$db = $this->getConnection();
$query = $db->prepare('UPDATE utente SET reset_token=? WHERE email=?');

$token = bin2hex(random_bytes(15));

$query->bindParam(1, $token);
$query->bindParam(2, $email);

$query->execute();

return $token;
```

Fatto ciò viene richiamato il model di gestione dell'email che, dopo aver impostato tutte le opzioni correttamente, invia l'email con il link alla pagina di reset (non ancora creata).

```
$body = "Per reimpostare la tua password premi sul seguente link: \n" . URL . "reset/
reset/" . $token;

try{
    $mail = new PHPMailer();
    ... opzioni ...
    $mail->SetFrom(EMAIL);
    $mail->Subject = "Grotti Ticino - Reimposta la tua password";
    $mail->Body = $body;
    $mail->AddAddress($email);

    if(!$mail->Send()) {
        $error = 'Mail error: ' . $mail->ErrorInfo;
        $_SESSION['warning'] = $error;
        header('Location: ' . URL . 'warning');
        exit();
    } else {
        return true;
    }
} catch (Exception $e) {
    $_SESSION['warning'] = $e->getCode() . " - " . $e->getMessage();
    header('Location: ' . URL . 'warning');
    exit();
}
```

Per fare tutto ciò ho dovuto generare un nuovo account gmail chiamato [grottiticino@gmail.com](mailto:grottiticino@gmail.com).

**Problemi riscontrati e soluzioni adottate**

Il problema che ho riscontrato oggi è che quando provo ad inviarmi un'email di reset della password il sito inizia a caricare all'infinito e non invia alcuna email però non genera nemmeno degli errori. Non sono riuscito a capire cosa e dove sia il problema.

**Punto della situazione rispetto alla pianificazione**

Sono in anticipo sulla pianificazione.

**Programma di massima per la prossima giornata di lavoro**

La prossima giornata voglio terminare il reset della password.

# Diario di lavoro

Luogo	Trevano
Data	15.10.2019

## Lavori svolti

Oggi durante le prime due ho risolto i problemi riscontrati la lezione precedente, esso era dovuto al fatto che la rete della scuola bloccava il traffico in uscita e entrata con il protocollo TTLS quindi il sito cercava di collegarsi all'infinito senza poterlo mai farlo. Sono venuto a conoscenza di questa cosa perché anche Filippo Finke stava avendo lo stesso problema ed ha scoperto che il traffico era bloccato. Abbiamo quindi chiesto al docente Raimondi di sbloccare il traffico del protocollo sulla porta 587 e il problema si è risolto. A questo punto ho iniziato a testare che tutto funzionasse correttamente.

Per poter cambiare le password ho dovuto creare una nuova funzione nella classe db\_connection.

```
public function setPassword($email, $password) {
    try{
        $db = $this->getConnection();
        $query = $db->prepare('UPDATE utente SET password=? WHERE email=?');

        $password = hash('sha256', $password);
        $query->bindParam(1, $password);
        $query->bindParam(2, $email);

        $query->execute();

    }catch (Exception $e){
        $_SESSION['warning'] = $e->getCode() . " - " . $e->getMessage();
        header('Location: ' . URL . 'warning');
        exit();
    }
}
```

Ed essa viene richiamata dal controller reset.

```
...
$user = (new DBConnection())->getUser($_SESSION['mail_sent']);

if($user['reset_token'] == $token) {

    (new DBConnection())->setPassword($_SESSION['mail_sent'], $_SESSION['password']);
    $_SESSION['password_change'] = true;
...
}
```

Fatto ciò, dato che ho ricevuto una chiave delle API di Google funzionante, ho potuto testare la pagina di aggiunta dei grotti che comprende il geocoding. Anche essa praticamente funzionava già ed ho quindi dovuto modificare solamente dei piccoli dettagli.

## Problemi riscontrati e soluzioni adottate

Oggi non ho riscontrato grossi problemi.

Punto della situazione rispetto alla pianificazione

Sono in anticipo sulla pianificazione.

Programma di massima per la prossima giornata di lavoro

La prossima giornata voglio iniziare la pagina di amministrazione.

# Diario di lavoro

Luogo	Trevano
Data	17.10.2019

## Lavori svolti

Oggi ho inizialmente messo a posto un problema che ho notato testando il sito ed era che le stelle delle valutazioni nella pagina dei grotti non erano selezionabili e quindi non si poteva votare. Questo era dovuto a delle classi CSS sbagliate ed ho risolto rimuovendo le classi inutili.

Fatto ciò sono passato a creare la pagina di admin, come prima cosa ho creato l'interfaccia di base della pagina che si presenta così.

The screenshot shows the homepage of the administration interface. At the top, there is a header bar with the text "Grotti Ticinesi" on the left and navigation links "Home", "Amministrazione", "Aggiungi Grotto", and "Logout" on the right. Below the header, the title "Amministrazione grotti Ticino" is centered, followed by the text "Connesso come: Matteo Forni (matteoforni)". There are three main buttons: "Gestione utenti" (with a blue "GESTISCI UTENTI" button), "Gestione Grotti" (with a blue "GESTISCI GROTTI" button), and "Gestione Inserimenti" (with a blue "GESTISCI INSERIMENTI" button). At the bottom of the page, a footer bar contains the text "© 2018 Copyright: Piattaforma per la scoperta dei grotti del Canton Ticino".

Quando si preme uno dei tre button appare la tabella relativa.

The screenshot shows the "Gestione utenti" (User Management) page. The title "Amministrazione grotti Ticino" and user information "Connesso come: Matteo Forni (matteoforni)" are at the top. Below is a table with columns: Email, Nome, Cognome, Username, Ruolo, Modifica (yellow button), and Elimina (red button). Two users are listed:

Email	Nome	Cognome	Username	Ruolo	Modifica	Elimina
matteo.forni@samtrevano.ch	Matteo	Forni	matteoforni	admin	<span>MODIFICA</span>	<span>ELIMINA</span>
matteoforni@hotmail.com	Matteo	Forni	fornimatteo	utente	<span>MODIFICA</span>	<span>ELIMINA</span>

Below the table, there are links for "Gestione Grotti" and "Gestione Inserimenti".

Le colonne della tabelle sono cliccabili e poteranno ad una pagina di gestione del singolo utente / grotto.

Per fare questa pagina ho dovuto modificare il metodo `getGrotti` dato che prima

ritornava solo i grotti verificati mentre ora io necessitavo anche quelli non ancora controllati.

```
public function getGrotti($validati){  
    try {  
        if($validati){  
            $verificato = 1;  
        }else{  
            $verificato = 0;  
        }  
    ...  
}
```

#### Problemi riscontrati e soluzioni adottate

Oggi l'unico problema che ho riscontrato era che i metodi getUtenti e getGrotti ritornavano direttamente la query senza farvi un fetch. Nelle pagine fatte in precedenza non vi erano problemi mentre oggi generava il seguente errore:

*Uncaught PDOException: You cannot serialize or unserialize PDOStatement instances*

Dopo un po di tentativi sono arrivato alla conclusione, anche grazie ad un aiuto di Filippo Finke che mi ha aiutato a isolare la causa del problema, di ritornare nelle due funzioni il fetch della query.

```
return $query->fetchAll();
```

#### Punto della situazione rispetto alla pianificazione

Sono in anticipo sulla pianificazione.

#### Programma di massima per la prossima giornata di lavoro

La prossima giornata voglio terminare la pagina di amministrazione, più precisamente le pagine di modifica / eliminazione di un'utente / grotto.

# Diario di lavoro

Luogo	Trevano
Data	18.10.2019

## Lavori svolti

Oggi ho iniziato ricontrollando la pagina di amministrazione e mi sono reso conto che era piuttosto scomoda da utilizzare con i bottoni che facevano apparire le tabelle, ho quindi deciso di rimuoverli e lasciare le tabelle visibili. Queste sono limitate a mostrare al massimo 7 righe per pagina così che chi visualizza la pagina non debba scrollare troppo per trovare il record che cerca.

Dopo aver ristrutturato il frontend ho continuato con quello che mi mancava, ho iniziato creando la funzione che richiama la pagina "gestione" che mostra in un form l'utente / grotto da modificare. La pagina è un insieme delle pagine di aggiunta di un grotto e di registrazione di un utente e andrà a richiamare i metodi che verificheranno le modifiche e le renderanno effettive (questi metodi sono ancora da fare). Fatto ciò sono passato a scrivere le funzioni di approvazione di un grotto e di eliminazione di un elemento sia nel controller che nel model.

A seguire prima la parte della funzione di eliminazione situata nel controller che verifica che tipo di campo si vuole eliminare, se questo esiste e, nel caso degli utenti, che ci sia sempre almeno un admin dopo l'eliminazione.

...

```
if($type == 'grotto') {
    $id = filter_var($im->checkInput($id), FILTER_SANITIZE_NUMBER_INT);
    $grotto = (new DBConnection)->getGrotto($id);
    if($grotto != null) {
        (new DBConnection)->delete($type, $id);
    }
} elseif ($type == 'utente') {
    $id = filter_var($im->checkInput($id), FILTER_SANITIZE_EMAIL);
    $utenti = (new DBConnection)->getUsers();
    $utente = (new DBConnection)->getUser($id);
    if($utente != null) {
        foreach($utenti as $item) {
            if($item['nome_ruolo'] == 'admin' && $item['id'] != $utente['id']) {
                (new DBConnection)->delete($type, $id);
            } else{
                array_push($errors, "Deve sempre esserci almeno un admin");
                $_SESSION['errors'] = $errors;
            }
        }
    }
}
...

```

Fatto ciò ho impiegato il tempo restante nella lezione per testare il corretto funzionamento del codice scritto.

## Problemi riscontrati e soluzioni adottate

Oggi ho riscontrato un problema durante l'eliminazione di un campo dalla tabella, infatti cercavo di eliminare un utente o un grotto ma in entrambi i casi non mi era

consentito dato che essi erano inseriti nella tabella **voto** come foreign key.  
Per risolvere il problema ho dovuto aggiungere al database dove dichiaro le chiavi esterne l'operazione da seguire in caso di eliminazione.

```
create table voto(
    email_utente varchar(50),
    id_grotto int,
    voto float not null,
    primary key(email_utente, id_grotto),
    foreign key(email_utente) references utente(email) on delete cascade,
    foreign key(id_grotto) references grotto(id) on delete cascade
);
```

Come si può vedere sopra la parte in grassetto fa sì che all'eliminazione del campo a cui si fa riferimento vengano eliminati tutti i figli. Quindi in questo caso succederà che:

- Se si elimina un grotto tutti i voti ad esso verranno persi
- Se si elimina un utente tutti i voti effettuati da esse verranno persi

Questo ho dovuto farlo anche nella tabella foto per i grotti.

Punto della situazione rispetto alla pianificazione

Sono in anticipo sulla pianificazione.

Programma di massima per la prossima giornata di lavoro

La prossima giornata voglio terminare la pagina di modifica di un utente / grotto.

# Diario di lavoro

Luogo	Trevano
Data	22.10.2019

## Lavori svolti

Oggi ho continuato come da programma con la pagina di modifica di un utente / grotto. Più precisamente ho sviluppato le due funzioni che verificano gli inserimenti dell'utente (admin) e gli inserisce nel db se corretti.

Le due funzioni sono piuttosto simili ma hanno alcuni particolari diversi dovuti al fatto che un utente deve essere sottoposto a più controlli di un grotto. Infatti la funzione per gli utenti verifica inizialmente che non sia stata modificata la password, se questo è successo la scrive nel database e poi invia un'email all'utente che ha subito la modifica per informarlo della sua nuova password.

Fatto ciò si verifica se gli altri campi sono diversi da come erano in precedenza, se ci sono differenze esse vengono inserite nel database. Un'ulteriore caratteristica della funzione per utenti è che se si cambia ruolo più precisamente da admin a utente si deve verificare che vi sia almeno un altro admin per rendere effettiva la modifica.

La funzione per grotti invece controlla solamente le differenze tra dati vecchi e nuovi e le scrive nel db. Ha però come caratteristica che se si modifica un campo dell'indirizzo bisogna ricalcolare le coordinate, questo è fatto ogni volta da frontend quando si esegue il submit del form. Per la modifica di un grotto si può anche visualizzare una mappa che mostra dove verrà messo il puntatore.

## Problemi riscontrati e soluzioni adottate

Oggi non ho riscontrato grandi problemi.

## Punto della situazione rispetto alla pianificazione

Sono in anticipo sulla pianificazione.

## Programma di massima per la prossima giornata di lavoro

La prossima giornata voglio terminare la pagina di modifica di un utente / grotto e iniziare a implementare la parte di aggiunta di un utente da parte di un admin.

# Diario di lavoro

Luogo	Trevano
Data	24.10.2019

## Lavori svolti

Oggi ho continuato come da programma con la pagina di modifica di un utente / grotto, dopo aver aggiunto i validatori frontend e aver corretto dei piccoli bug grafici l'ho terminata ed ho potuto testarla.

Fatto ciò sono passato a sviluppare la pagina di aggiunta di un utente da parte di un admin, la pagina è praticamente uguale a quella di registrazione le uniche differenze sono che non si può inserire una password perché essa verrà creata in maniera random ed inviata via email all'indirizzo inserito nella creazione dell'account.

Fatto ciò ho continuato la documentazione nel capitolo implementazione e facendo ciò mi sono reso conto che mancavano delle parti più o meno importanti nel sito di cui mi ero dimenticato e le ho inserite nel file TODO.md.

Dopodiché ho iniziato a testare le validazioni frontend per essere certo che tutte verifichino tutto correttamente.

## Problemi riscontrati e soluzioni adottate

Oggi non ho riscontrato grandi problemi.

## Punto della situazione rispetto alla pianificazione

Sono in anticipo sulla pianificazione.

## Programma di massima per la prossima giornata di lavoro

La prossima giornata voglio implementare la possibilità di vedere il numero di valutazioni effettuate su di un determinato grotto e la finestra di modifica della password al primo accesso di un utente creato da un admin.

# Diario di lavoro

Luogo	Trevano
Data	25.10.2019

## Lavori svolti

Oggi come da programma ho implementato il codice che consente di visualizzare il numero di valutazioni eseguite su un grotto. Esso è piuttosto semplice ed è composto da una stampa di un valore ricavato dal database con la seguente query:

```
SELECT count(*) FROM voto where id_grotto=?
```

Fatto ciò ho creato la pagina di cambio password al primo login. Essa viene richiamata dopo il login e verifica che il campo first\_login sia impostato a 1, se lo è richiede all'utente di inserire una nuova password e poi lo rimanda ad eseguire nuovamente il login mentre se è a 0 fa loggare l'utente normalmente.

Essa è abbastanza simile alla pagina di reset della password dato che i campi sono gli stessi tranne che non bisogna verificare che l'email sia in uso ed essa non deve poter nemmeno essere modificata.

Se tutto è corretto (le password ripetute sono uguali) viene impostata la nuova password e viene cambiato il campo first\_login da 1 a 0.

```
(new DBConnection)->setPassword($email, $password);  
(new DBConnection)->setFirstLogin($email);
```

Dopodiché ho sviluppato in parte l'aggiunta delle immagini. Per capire come fare mi sono basato sull'esempio del seguente sito:

<https://code.tutsplus.com/tutorials/how-to-upload-a-file-in-php-with-example--cms-31763>

La funzionalità è stata aggiunta nella pagina che mostra il grotto selezionato affianco il form che consente di votare.

Una volta inserita l'immagine si effettuano i vari controlli (se è un'immagine,...) e se li supera tutti viene salvata (per evitare problemi con spazi/caratteri speciali,...) con un nome composto dall'hash del contenuto dell'immagine più l'estensione, grazie a questo nome si può anche verificare che l'immagine non esista già.

**Problemi riscontrati e soluzioni adottate**

Oggi il problema era che non avevo mai utilizzato i files e l'upload di files con PHP quindi ho dovuto ricercare ed informarmi sulle tecniche da utilizzare. Questo lo sono riuscito a fare grazie al sito menzionato in precedenza ed alcuni consigli da parte di compagni sul nome dei files da utilizzare.

**Punto della situazione rispetto alla pianificazione**

Sono in anticipo sulla pianificazione.

**Programma di massima per la prossima giornata di lavoro**

La prossima giornata voglio terminare l'aggiunta delle immagini ed occuparmi della documentazione.

# Diario di lavoro

Luogo	Trevano
Data	05.11.2019

## Lavori svolti

Oggi come da programma ho migliorato e completato il codice che consente di associare delle immagini ad un grotto. Ho dovuto, per farlo funzionare, modificare il percorso in cui salvavo le immagini dato che lo avevo inserito sbagliato e sviluppare la parte di aggiunta al database.

Dopo aver eseguito dei test del funzionamento ho creato la sezione nella pagina admin in cui un utente con i privilegi necessari può eliminare delle immagini. Questa parte è piuttosto simile alle altre tabelle dato che i dati vengono mostrati nello stesso modo. Nella parte frontend l'unica cosa più complicata è la stampa del nome di un grotto in base all'id di esso, questo viene fatto con il seguente codice:

```
echo $_SESSION['grotti_validati'][array_search($row['grotto'], array_column($_SESSION['grotti_validati'], 'id'))]['nome'];
```

La parte seguente serve a cercare quale elemento ha nella colonna id il valore salvato nella variabile \$row['grotto'] nella sessione contenente i grotti validati così da stamparne il nome.

```
array_search($row['grotto'], array_column($_SESSION['grotti_validati'], 'id'))
```

Fatto ciò ho iniziato a scrivere la parte backend dell'eliminazione, per farlo ho modificato le funzioni già usate per eliminare un grotto/utente.

## Problemi riscontrati e soluzioni adottate

Oggi il problema che ho riscontrato era che non capivo perché dopo aver eliminato un'immagine non potessi associarla nuovamente al grotto e mi sono reso conto che eliminavo l'immagine solamente dal database e non dalla cartella dove era salvata. Per risolvere il problema mi è bastato aggiungere un costrutto try catch contenente sia l'eliminazione dal db che quella dal filesystem. Per toglierla dal filesystem uso il metodo unlink.

```
try {
    unlink($immagine['path']);
    (new DBConnection)->delete($type, $id);
} catch (Exception $e) {
    ...
}
```

## Punto della situazione rispetto alla pianificazione

Sono in anticipo sulla pianificazione.

Programma di massima per la prossima giornata di lavoro

La prossima giornata voglio occuparmi della documentazione e mettere a posto dei piccoli errori noti nel sito.

# Diario di lavoro

Luogo	Trevano
Data	07.11.2019

## Lavori svolti

Oggi come da programma ho continuato la documentazione, più precisamente il capitolo dell'implementazione terminando la spiegazione del database e iniziando quella della pagina principale.

Fatto ciò ho cercato di risolvere alcuni problemi nella pagina in particolare il fatto che le immagini non venivano eliminate dal filesystem quando veniva rimosso un grotto da un admin. Per risolvere il problema ho aggiunto il codice che verifica, prima dell'eliminazione che una località, se essa ha immagini collegate e se vi sono le cancella tramite il metodo unlink(path) di PHP. Per fare funzionare tutto ciò, dato che uso linux, ho però dovuto modificare a livello di filesystem il proprietario della cartella *immagini* cambiandolo con l'utente apposito di apache chiamato www-data e con il gruppo chiamato allo stesso modo.

Quando sono riuscito a risolvere il problema precedente ho verificato parte dei form dato che avevo notato che in alcuni mancavano dei controlli.

## Problemi riscontrati e soluzioni adottate

Oggi non ho riscontrato problemi.

## Punto della situazione rispetto alla pianificazione

Sono in anticipo sulla pianificazione.

## Programma di massima per la prossima giornata di lavoro

La prossima giornata voglio occuparmi della documentazione e migliorare delle piccole parti del sito.

# Diario di lavoro

Luogo	Trevano
Data	08.11.2019

## Lavori svolti

Oggi come da programma, nelle prime due ore ho continuato la documentazione, più precisamente il capitolo dell'implementazione completando il sottocapitolo dedicato alla pagina home e scrivendo quelli riguardanti la pagina che mostra un grotto in dettaglio, quella di login, quella di registrazione e la pagina di reset della email.

Documentando la pagina di reset della email mi sono reso conto che era stata creata senza pensare accuratamente alla situazione, essa inizialmente richiedeva l'email dell'account e direttamente la nuova password, fatto ciò inviava un token per email (che salvava anche nel db) e se il token del link, inviato per email, premuto dall'utente corrispondeva con quello nel database cambiava la password.

Dopo un'analisi più attenta ho ristrutturato la pagina facendo sì che inizialmente chieda solo l'email e dopo aver inviato il messaggio con il token, se corrisponde con quello nel database, mostra il form per cambiare password. Inoltre adesso tutta la pagina funziona con i cookies e non con le sessioni così che se si chiude il browser durante le operazioni non si perde tutto.

## Problemi riscontrati e soluzioni adottate

Oggi ho avuto parecchie problemi con i cookies dato che non li avevo mai utilizzati, inizialmente li creavo solo per una pagina e sulle altre non funzionavano e ho scoperto che per utilizzarli in modo globale quando li si crea bisogna inserire anche la path di dove devono essere utilizzati, in questo caso '/' per usarli ovunque:

```
setcookie(nome, valore, tempo di durata, '/');
```

Il secondo problema che ho riscontrato era che quando eliminavo un cookie da una pagina esso si eliminava ma se cambiavo pagina si ricreava automaticamente. Dopo diverso tempo perso ho scoperto che era un problema di chrome che lo teneva salvato e lo ricaricava in continuazione. Ho risolto eliminando i cookies dalle impostazioni del browser.

## Punto della situazione rispetto alla pianificazione

Sono in anticipo sulla pianificazione.

## Programma di massima per la prossima giornata di lavoro

La prossima giornata voglio occuparmi della documentazione e migliorare delle piccole parti del sito.

# Diario di lavoro

---

Luogo	Trevano
Data	12.11.2019

## Lavori svolti

Oggi come da programma, nelle prime due ore ho continuato la documentazione, più precisamente il capitolo dell'implementazione completando il sottocapitolo dedicato alla pagina di amministrazione e di aggiunta di un grotto.

Fatto ciò ho messo a posto i models dato che molte funzioni erano nella classe DBConnection quando avrebbero dovuto essere separate nei model relativi alla tabella. Dopo aver spostato i metodi nel posto corretto ho verificato che tutto funzionasse ancora correttamente.

Infine ho aggiunto la libreria che esegue lo smooth scroll nella pagina di amministrazione, essa si chiama smoothie-js e la documentazione e il codice possono essere trovati al seguente link:

<https://github.com/marlospomin/smoothie>

## Problemi riscontrati e soluzioni adottate

Oggi non ho avuto problemi.

## Punto della situazione rispetto alla pianificazione

Sono in anticipo sulla pianificazione.

## Programma di massima per la prossima giornata di lavoro

La prossima giornata voglio occuparmi della documentazione e iniziare a creare lo schema UML delle classi.

# Diario di lavoro

---

Luogo	Trevano
Data	14.11.2019

## Lavori svolti

Oggi come da programma, nelle prime due ore ho creato i diagrammi UML delle classi per ogni cartella contenente file PHP nel progetto. Ho fatto così perché generandone solamente uno sarebbe risultato uno schema molto complesso. Dopodiché ho continuato la documentazione, più precisamente il capitolo dell'implementazione completando il sottocapitolo che parla dei models inserendo appunto anche lo schema relativo creato in precedenza.

## Problemi riscontrati e soluzioni adottate

Oggi non ho avuto problemi.

## Punto della situazione rispetto alla pianificazione

Sono in anticipo sulla pianificazione.

## Programma di massima per la prossima giornata di lavoro

La prossima giornata voglio occuparmi della documentazione e cercare di implementare la condivisione di un grotto sui social.

# Diario di lavoro

Luogo	Trevano
Data	15.11.2019

## Lavori svolti

Oggi ho iniziato la lezione documentandomi su come poter condividere un link o un'immagine di un grotto su Instagram ma ho scoperto leggendo la documentazione ufficiale che questo non è possibile ed è consentito solo ad applicazioni mobile.

Questo è il link della documentazione ufficiale:

<https://developers.facebook.com/docs/instagram-basic-display-api/>

Dopodiché mi sono documentato su come fare per condividere i link su facebook. Su questa piattaforma è possibile ma il sito deve essere su un hosting. Ho provato quindi a caricarlo sul hosting della scuola (infomaniak) ma non consente di caricare cartelle e quindi dovrei caricare un file per volta e ci metterei troppo. Ho quindi provato a connettermi con filezilla ma il proxy di scuola blocca le connessioni e quindi dovrò provare da casa. Inoltre ho scoperto che l'hosting di infomaniak non consente di usare i trigger di MySQL e quindi i voti sui grotti non funzionerebbero.

Fatto ciò mi sono occupato di abbellire la pagina di amministrazione eliminando i bottoni dalle tabelle e sostituendoli con delle icone prese da font awesome.

<https://fontawesome.com/icons?d=gallery>.

In seguito mi sono occupato di verificare il funzionamento corretto di tutti i forms (click senza niente, click con dei campi mancanti,...) e del corretto caricamento del header in base all'utente utilizzato (se sono admin devo vedere anche la pagina di amministrazione altrimenti no e se non sono loggato devo vedere solo la pagina di login).

Infine ho continuato con la documentazione (capitolo implementazione).

## Problemi riscontrati e soluzioni adottate

Oggi non ho avuto problemi.

## Punto della situazione rispetto alla pianificazione

Sono in orario.

## Programma di massima per la prossima giornata di lavoro

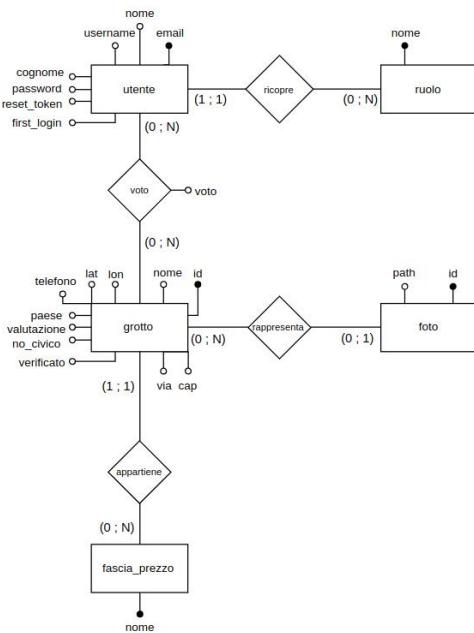
La prossima giornata voglio occuparmi della documentazione e del ER.

# Diario di lavoro

Luogo	Trevano
Data	19.11.2019

## Lavori svolti

Oggi ho iniziato la lezione creando il diagramma ER conclusivo ed il relativo schema logico. Essi hanno come differenze dai precedenti che contengono più campi che non erano stati previsti all'inizio. Di seguito i due schemi aggiornati.



RUOLO(nome)

UTENTE(email, nome, cognome, username, password, reset\_token, first\_login\*, nome\_ruolo(FK))

FASCIA\_PREZZO(nome)

GROTTA(id, via, paese, cap, no\_civico, lat, lon, nome, valutazione\*, telefono, verificato, fascia\_prezzo(FK))

FOTO(id, path, id\_grotto\*(FK)) path unique

VOTO(email\_utente(FK), id\_grotto(fk), voto)

Fatto ciò ho continuato con la documentazione terminando il capitolo relativo alle classi nella cartella models ed ho iniziato quello sui controllers.

## Problemi riscontrati e soluzioni adottate

Oggi non ho avuto problemi.

## Punto della situazione rispetto alla pianificazione

Sono in orario.

Piattaforma per la scoperta dei Grotti del Canton Ticino

1/2

Programma di massima per la prossima giornata di lavoro

La prossima giornata voglio occuparmi della documentazione e fare un po' di testing.

# Diario di lavoro

Luogo	Trevano
Data	21.11.2019

## Lavori svolti

Oggi ho iniziato la lezione continuando la documentazione portando quasi a termine il capitolo dei controller. Dopodiché ho iniziato a fare un po di testing e mi sono accorto che era scomodo per un utente non vedere mai con che utente è connesso e che tipi di privilegi ha, per risolvere ho aggiunto nella navbar il nome e cognome dello user ed andandoci sopra con il mouse si vede il tipo di utente (utente o admin). Questa cosa non viene mostrata su dispositivi mobili di dimensioni dei telefoni.

Fatto ciò mi sono accorto che nella pagina di amministrazione non chiedevo conferma prima di eliminare un campo ed ho quindi implementato il codice che lo fa con dei modali.

## Problemi riscontrati e soluzioni adottate

Oggi non ho avuto problemi.

## Punto della situazione rispetto alla pianificazione

Sono in orario.

## Programma di massima per la prossima giornata di lavoro

La prossima giornata voglio occuparmi della documentazione e continuare con il testing.

# Diario di lavoro

Luogo	Trevano
Data	22.11.2019

## Lavori svolti

Oggi ho iniziato la lezione continuando la documentazione terminando il capitolo dei controller e con esso la sezione dell'implementazione.

Fatto ciò mi sono dedicato a del testing generale del progetto, ho riscontrato tre errori principali che erano i seguenti:

- L'admin non poteva eliminare gli utenti perché otteneva sempre il messaggio d'errore “non puoi eliminare il tuo stesso account”
- Nella mappa le valutazioni non venivano caricate
- Dopo l'aggiunta di un grotto rimaneva, sotto il messaggio di conferma, la mappa per verificare la posizione del ristorante.
- Se si fa una richiesta da remoto senza passare dal controller (es. curl) vengono saltati tutti i controlli perché vengono eseguiti solo nei controllers e non nei models.
- La prima lettera del tipo di utente nella toolbar era minuscola e non bella da vedere.

Per risolvere il primo errore ho impiegato parecchio tempo perché non trovavo dove fosse il problema ma alla fine ho scoperto che c'era, in un if, un controllo scritto al contrario. Era scritto `=!` e non `!=`.

Il secondo bug l'ho risolto con un workaround dal momento che non si riesce ad eseguire le funzioni della libreria auxiliary-rater nelle finestre all'interno della mappa. Ho quindi optato per mostrare il numero e non le stelle quindi se vi sono voti stampa “Valutazione: 3 su 5” mentre se non ve ne sono mostra “Nessun voto”.

Per non mostrare la mappa l'ho solamente spostata all'interno del form dove si inseriscono i dati.

Il quarto bug è ancora da risolvere ma dovrò fare, nei models, delle verifiche con delle regex.

Infine l'ultimo bug trovato l'ho risolto utilizzando il metodo `ucfirst(string)` durante la stampa.

## Problemi riscontrati e soluzioni adottate

Oggi l'unico problema che ho avuto era trovare cosa causasse il primo bug della lista sopra citata. Ho impiegato quasi 40 minuti ma l'ho trovato.

Punto della situazione rispetto alla pianificazione

Sono in orario.

Programma di massima per la prossima giornata di lavoro

La prossima giornata voglio iniziare il capitolo dei test nella documentazione e risolvere il problema 4 della lista sopra.

# Diario di lavoro

Luogo	Trevano
Data	26.11.2019

## Lavori svolti

Oggi ho iniziato la lezione eseguendo dei test e mi sono reso conto che all'eliminazione di un utente un admin non riceveva alcun messaggio di successo e quindi li ho implementati utilizzando gli Alerts di Material Design Bootstrap.

Fatto ciò ho notato che le medie delle votazioni erano in alcuni casi sbagliate perché non arrotondate allo 0.5 e quindi l'ho implementato modificando il codice come segue.

```
...
set @avg_review = (select avg(voto) from voto where id_grotto=NEW.id_grotto);
update grotto set valutazione = @avg_review where id=NEW.id_grotto;
...

...
set @avg_review = (select avg(voto) from voto where id_grotto=NEW.id_grotto);
set @avg_round = round(@avg_review*2)/2;
update grotto set valutazione = @avg_round where id=NEW.id_grotto;
...
```

In questo modo ottengo sempre il valore arrotondato allo 0.5 più vicino.

Dopodiché ho risolto alcuni problemi di sessioni che non chiudevo e che causavano alcuni problemi nel caricare i dati semplicemente andando alla fine del loro utilizzo e inserendo il codice: `unset($_SESSION['nome della sessione']);`

Fatto ciò mi sono occupato di trasformare le scritte delle fasce di prezzo in simboli traducendole nel seguente modo:

- Caro --> \$\$\$
- Nella norma --> \$\$
- Buon prezzo --> \$

Per farlo ho utilizzato le API di font awesome come per tutte le altre icone.

Infine, dopo il colloquio con il docente responsabile per fare il punto della situazione, ho continuato la documentazione iniziando a fare i primi test case.

## Problemi riscontrati e soluzioni adottate

Oggi non ho avuto grossi problemi.

Punto della situazione rispetto alla pianificazione

Sono in orario.

Programma di massima per la prossima giornata di lavoro

La prossima giornata voglio continuare il capitolo dei test nella documentazione e continuare con il testing.

# Diario di lavoro

---

Luogo	Trevano
Data	28.11.2019

## Lavori svolti

Oggi ho iniziato la lezione sviluppando due nuovi test case che verificano il corretto funzionamento della pagina di login, di modifica della password e del primo login. Eseguendo i test scritti ho notato alcuni problemi nella validazione dei campi, avevo dimenticato alcune validazioni e altre non erano fatte con cura. Ho dunque passato il resto della lezione a risolvere i problemi riscontrati e la prossima volta dovrò testare che le modifiche apportate funzionino e non creino problemi ad altre parti del sito.

## Problemi riscontrati e soluzioni adottate

Oggi non ho avuto grossi problemi tranne nel testare le regex che verificano i campi dal momento che non sono un esperto nel loro utilizzo. Ho dunque dovuto informarmi su internet.

## Punto della situazione rispetto alla pianificazione

Sono in orario.

## Programma di massima per la prossima giornata di lavoro

La prossima giornata voglio continuare il capitolo dei test nella documentazione e continuare con il testing.

# Diario di lavoro

Luogo	Trevano
Data	29.11.2019

## Lavori svolti

Oggi ho iniziato la lezione sviluppando tutti i test case mancanti così da completare il capitolo dei protocolli di test.

Fatto ciò ho fatto alcuni test sui form di aggiunta degli utenti da parte dell'admin ed ho apportato alcune modifiche al contenuto della mail contenente la password che viene inviata all'account appena creato aggiungendo il link alla pagina di login.

Dopodiché ho verificato che i validatori frontend venissero caricati su tutti i form del sito ed ho iniziato a testarne il funzionamento (sia frontend che quelli backend) seguendo i miei protocolli di test appena generati.

Fatto ciò ho fatto un test e mi sono reso conto che eliminando un utente elimino anche i suoi voti ma non aggiorno la tabella se non vi sono più voti. Quindi ho creato un nuovo trigger che, se non vi sono più voti su un grotto, setta a null il campo valutazione della località.

```
delimiter $$$
create trigger avg_after_delete after delete on voto for each row
begin
    set @counter = (select count(*) from voto where id_grotto=OLD.id_grotto);
    IF (@counter = 0) THEN
        update grotto set valutazione=null where id=OLD.id_grotto;
    END IF;
end;
$$$
delimiter ;
```

Infine ho aggiunto a tutti i validatori backend la regola che accetta i caratteri accentati perché mi sono accorto testando la pagina di registrazione che non li accettavo.

## Problemi riscontrati e soluzioni adottate

Oggi non ho avuto grossi problemi.

## Punto della situazione rispetto alla pianificazione

Sono in orario.

## Programma di massima per la prossima giornata di lavoro

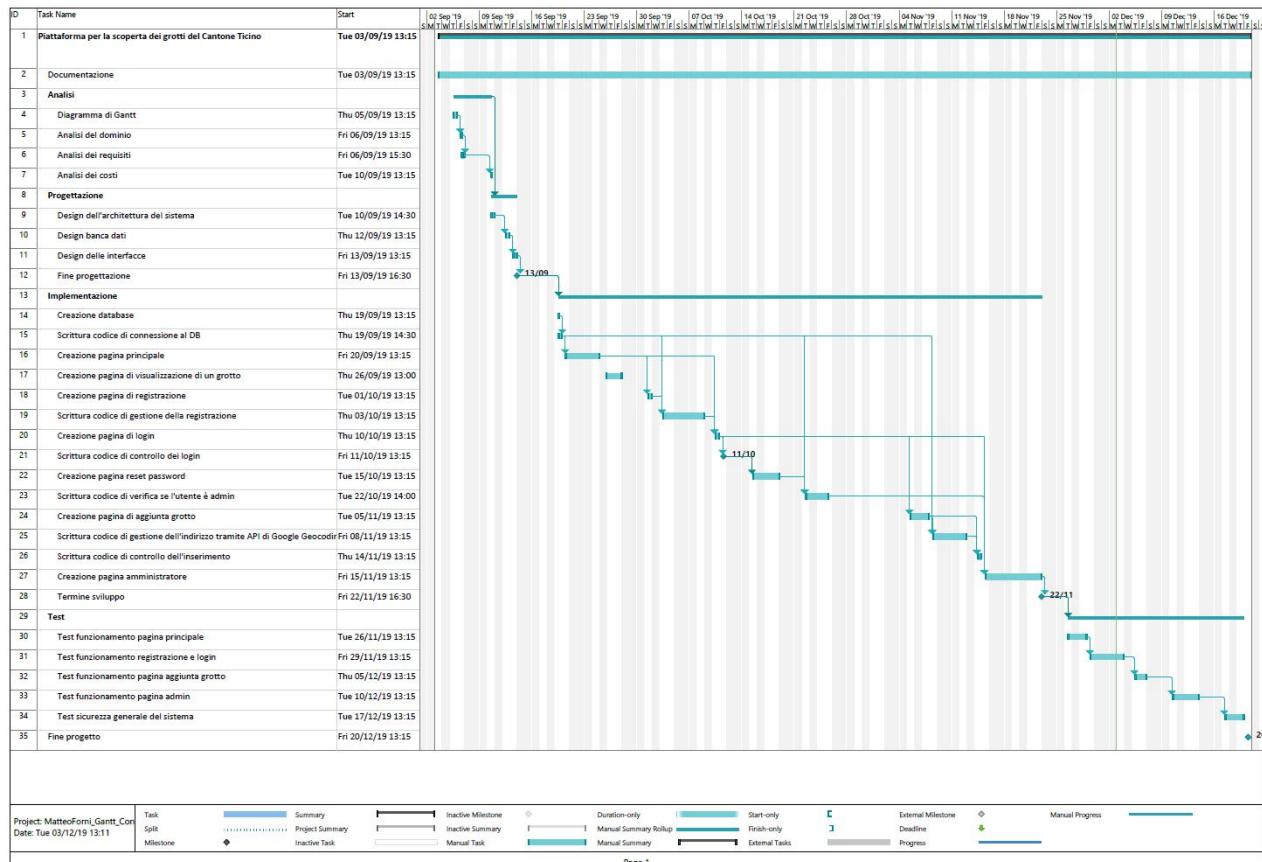
La prossima giornata voglio iniziare il capitolo dei risultati dei test nella documentazione e continuare con il testing.

# Diario di lavoro

Luogo	Trevano
Data	03.12.2019

## Lavori svolti

Oggi ho iniziato la lezione eseguendo tutti i test spiegati nel capitolo "Protocolli di test" e ho inserito i risultati nella tabella creata nel capitolo "Risultati dei test". Fatto ciò ho creato il gantt consuntivo.



Infine ho dovuto modificare leggermente il database per far sì che non venissero eliminati i voti all'eliminazione di un utente. Questo è stato fatto modificando la tabella voto.

Adesso la tabella è così (evidenziate sono le modifiche):

```
create table voto(
    id int primary key auto_increment,
    email_utente varchar(50),
    id_grotto int,
    voto float not null,
    foreign key(email_utente) references utente(email) on delete set null,
    foreign key(id_grotto) references grotto(id) on delete cascade
);
```

Piattaforma per la scoperta dei Grotti del Canton Ticino

1/2

**Problemi riscontrati e soluzioni adottate**

Oggi non ho avuto grossi problemi.

**Punto della situazione rispetto alla pianificazione**

Sono in orario.

**Programma di massima per la prossima giornata di lavoro**

La prossima giornata voglio continuare la documentazione e continuare con il testing.

# Diario di lavoro

---

Luogo	Trevano
Data	06.12.2019

## Lavori svolti

La lezione del 05.12 non ho lavorato perché siamo stati ospiti della HEP-FR per vedere la scuola e i corsi di bachelor da loro.

Oggi ho completato i capitoli del consuntivo e delle conclusioni della documentazione.

Inoltre ho commentato tutte le classi e i metodi che non erano già stati commentati in precedenza.

## Problemi riscontrati e soluzioni adottate

Oggi non ho avuto problemi.

## Punto della situazione rispetto alla pianificazione

Sono in orario.

## Programma di massima per la prossima giornata di lavoro

La prossima giornata voglio continuare la documentazione scrivendo l'abstract e eseguendo dei test generali di funzionamento e sicurezza.

# Diario di lavoro

---

Luogo	Trevano
Data	10.12.2019

## Lavori svolti

Oggi ho scritto l'abstract in italiano (un foglio a parte da inserire prima della documentazione) dopodiché ho riletto la documentazione in cerca di errori e ho fatto un po di testing al sito.

## Problemi riscontrati e soluzioni adottate

Oggi non ho avuto problemi.

## Punto della situazione rispetto alla pianificazione

Sono in orario.

## Programma di massima per la prossima giornata di lavoro

La prossima giornata voglio continuare la documentazione scrivendo l'abstract e eseguendo dei test generali di funzionamento.

# Diario di lavoro

---

Luogo	Trevano
Data	12.12.2019

**Lavori svolti**

Oggi ho scritto l'abstract in inglese dopodiché ho eseguito un pò di test generali per verificare di non aver dimenticato niente e non avere in giro piccoli errori.

**Problemi riscontrati e soluzioni adottate**

Oggi non ho avuto problemi.

**Punto della situazione rispetto alla pianificazione**

Sono in orario.

**Programma di massima per la prossima giornata di lavoro**

La prossima giornata voglio terminare la documentazione e rileggerla per verificare che non ci siano errori.

# Diario di lavoro

---

Luogo	Trevano
Data	13.12.2019

**Lavori svolti**

Oggi ho riletto la documentazione correggendo eventuali errori di scrittura trovati.

**Problemi riscontrati e soluzioni adottate**

Oggi non ho avuto problemi.

**Punto della situazione rispetto alla pianificazione**

Sono in orario.

**Programma di massima per la prossima giornata di lavoro**

La prossima giornata voglio stampare tutti i documenti ed iniziare a rilegarli.

# Diario di lavoro

---

Luogo	Trevano
Data	17.12.2019

**Lavori svolti**

Oggi ho stampato i file da consegnare al termine del progetto ed ho iniziato la rilegatura ma non sono riuscito a finirla dato che c'è una sola rilegatrice e tutti devono usarla.

**Problemi riscontrati e soluzioni adottate**

Oggi non ho avuto problemi.

**Punto della situazione rispetto alla pianificazione**

Sono in orario.

**Programma di massima per la prossima giornata di lavoro**

La prossima giornata voglio terminare la rilegatura.

# Diario di lavoro

---

Luogo	Trevano
Data	19.12.2019

**Lavori svolti**

Oggi ho terminato la rilegatura ed ho preparato il CD per la consegna.

**Problemi riscontrati e soluzioni adottate**

Oggi non ho avuto problemi.

**Punto della situazione rispetto alla pianificazione**

Sono in orario.

**Programma di massima per la prossima giornata di lavoro**

La prossima giornata voglio consegnare il progetto.

# Diario di lavoro

---

Luogo	Trevano
Data	20.12.2019

**Lavori svolti**

Oggi ho consegnato il progetto.

**Problemi riscontrati e soluzioni adottate**

Oggi non ho avuto problemi.

**Punto della situazione rispetto alla pianificazione**

Sono in orario.

**Programma di massima per la prossima giornata di lavoro**

-